UniCEUB – Centro Universitário de Brasília
FAET – Faculdade de Ciências Exatas e Tecnologia
Curso de Engenharia da Computação
Projeto Final

Simulação de Criptografia Quântica

Aluno: Cid Antunes Horta Júnior - Ra: 20317374
Orientador: Professor Thiago Toribio

Brasília, DF – Novembro de.2006

UniCEUB – Centro Universitário de Brasília
FAET – Faculdade de Ciências Exatas e Tecnologia
Curso de Engenharia da Computação
Projeto Final

Simulação de Criptografia Quântica

por

Cid A. Horta Jr
Ra: 20317374

Trabalho Final de Graduação

Professor Thiago Toribio
Orientador

Índice

Índice de Figuras

Índice de Tabelas

Resumo

Este trabalho apresentará um sistema criptográfico baseado na codificação de informação em estados quânticos para geração e distribuição de chaves criptográficas, que vem sendo proposta como uma alternativa aos sistemas criptográficos atuais. Esse sistema funciona por meio da distribuição de chaves criptográficas Quânticas, também conhecidas pela sigla “QKD” (*Quantum Key Distribution* – Distribuição Quântica de Chaves), procedimento que garante a inviolabilidade do compartilhamento da chave, haja vista a validade das leis da Física Quântica. Além da pesquisa, será desenvolvido um protótipo que simulará uma transmissão Quântica e a geração das chaves criptográficas.

O estudo foi dividido nos seguintes capítulos: Introdução, apresentação do tema; Referencial Teórico, pesquisa sobre os temas que dão suporte ao objetivo do trabalho; Método/Metodologia, onde serão comentados os passos seguidos na simulação proposta; e, por fim, os Resultados e Discussões, no qual serão relatados e analisados os resultados obtidos com o desfecho do trabalho. Dentro do referencial teórico, são apresentadas duas ciências que estão fortemente ligadas a Criptografia Quântica: Física Quântica e Criptografia.

A Física Quântica será abordada com a descrição das suas principais características e conceitos, além de sua aplicação dentro da Criptografia Quântica. A criptologia será estudada através da criptografia, uma subdivisão desta área. Serão descritos seus conceitos, de algumas de suas técnicas, seus modelos e algoritmos. Os principais protocolos de Criptografia Quântica também são abordados e um deles será utilizado para a simulação. Como conclusão, serão apresentados os resultados obtidos do simulador, que mostrou como a Criptografia Quântica, através da utilização de um de seus protocolos, pode agregar segurança no momento mais crítico de todo sistema criptográfico, o compartilhamento de chave de criptografia.

Palavras-Chave: Criptografia Quântica, Distribuição Quântica de Chaves.

Abstract

This work will present a cryptographic system based on data codification on quantum states in order to generate and distribute cryptographic keys. This codification is being proposed as an alternative to the present cryptographic systems. It works by the Quantum Key Distribution ("QKD"), procedure that warranties the inviolability of key sharing, based on the validity of the Quantum Physics laws. Besides the research, it will be presented here the prototype developed to simulate a quantum key generation.

The work is divided in the following chapters: introduction, the presentation of the subject; theoretical references, research on the subjects that support the work; methods/methodology, where the steps taken in the proposed simulation will be commented; and results and discussion, where the results of the work will be analyzed as a closure.

The approach to Quantum Physics will describe its main characteristics and concepts, its use on the Quantum Cryptography. Cryptology will be studied through the Cryptography. The main protocols of the Quantum Cryptography will also be approached, and one of them will be used in the simulation. The results of the simulation summed up as a conclusion have shown how Quantum Cryptography (through the use of one of the protocols) can aggregate safety at the most critical moment of the entire cryptograpic system, the sharing of a cryptographic key.

The theoretical referential will present the two sciences that are strongly related to the Quantum Cryptography: Quantum Physics and Cryptography.

Keyword: Quantum Cryptography, Quantum Key Distribution.

1. Introdução

Negócios milionários são realizados todos os dias por grandes empresas multinacionais via meio eletrônico, principalmente por e-mail. Até mesmo um simples indivíduo que tem acesso à Internet realiza operações comerciais e pessoais por meio desta. Entretanto, imagine o que aconteceria quando os dados que estão sendo emitidos por Alice (pessoa fictícia) com o propósito de garantir uma grande compra com Bob são decodificados e decifrados por Eva, um estranho. Esta responde a mensagem interceptada como se fosse Bob e concretiza o negócio, só que quem vai receber o dinheiro é a própria, e não o seu destinatário final.

O exemplo citado acima é uma pequena amostra de crimes virtuais que estão acontecendo a cada minuto em todo o mundo. Na tentativa de impedí-los, estudos estão sendo desenvolvidos com a finalidade de criar ambientes ágeis e seguros de transmissão de dados. Um deles utiliza chaves criptográficas para concretizar estas transações, uma vez que elas correspondem ao elo codificador de uma mensagem emitida e sua correta recepção. Ou seja, são responsáveis por codificar e decodificar os dados emitidos entre pessoas ou servidores.

Ao elaborar este projeto de monografia, busca-se delineá-lo dentro do contexto da Internet, redes de comunicação e chaves criptográficas. A partir daí, percebe-se o quanto é constante o problema narrado e como as tecnologias atuais estão vulneráveis aos avanços tecnológicos.

Com o objetivo de tentar apontar uma nova solução para este impasse, serão utilizadas áreas separadas de estudos, como a Criptografia, a Física Quântica e Criptografia Quântica, para se estruturar o protótipo base desse projeto. Entretanto, antes de unir estas ciências, serão utilizados os dois primeiros capítulos desta monografia para esboçá-las separadamente e contextualizá-las com o assunto em questão.

A criptografia, como será visto nos próximos capítulos, é responsável por “embaralhar” a mensagem quando emitida e “desembaralhar” quando recebida

pelo correto destinatário final, utilizando-se uma ou mais chaves específicas. Uma simples definição que esconde um complexo processo de produção, que será demonstrado na montagem do protótipo. Além disso, será demonstrado numa seção dedicada a esta matéria suas principais características e aplicações nos avanços das novas chaves criptográficas.

A Física Quântica estuda a natureza e seus fenômenos, métodos, fundamentos teóricos e objetivos epistemológicos em escalas de distâncias microscópicas. Ela ajuda a isolar um ambiente e, posteriormente, analisá-lo detalhadamente. Assim, quando se criar um ambiente seguro para realizar a transmissão de um pacote de dados, pode-se identificar quais alterações que deverão ser verificadas para se ter o conhecimento se a mensagem não foi decodificada por um agente estranho a relação.

O estudo da Criptografia Quântica demonstrará como é possível a comunicação para troca de chaves criptográficas de uma maneira segura dentro de um ambiente com maior nível de segurança que os meios de comunicações públicos, onde trafegam apenas as mensagens codificadas e não mais a chaves. Dessa forma, a Internet e outros meios eletrônicos de comunicação estarão mais seguros e serão mais eficientes. Ou seja, um usuário não sentirá medo ao utilizar a Internet para realizar um pagamento no banco ou usar seu cartão de crédito para pagar uma conta, por exemplo.

Para se trazer um pouco da teoria para a prática, será utilizada a união dos princípios da Física Quântica e da Criptografia para se ampliar o estudo da Criptografia Quântica e, com isso, desenvolver um protótipo que demonstrará o seu funcionamento por meio de um protocolo quântico. Uma demonstração de algo que só estudamos nos livros: o desempenho da Criptografia Quântica para a proteção de informações ou pacote de dados.

2. Referencial Teórico

2.1. Criptografia

Este capítulo tem por objetivo apresentar os conceitos de criptografia que estão relacionados, direta ou indiretamente, com o assunto deste trabalho, a Criptografia Quântica.

Além destes conceitos, será necessário, para melhor elucidar o objetivo a ser alcançado, descrever as principais técnicas de criptografia utilizadas atualmente e suas características. Importante ressaltar que este capítulo tem como propósito a construção de uma base teórica que irá ajudar no desenvolvimento e funcionamento do protótipo. A partir disto, vale dizer que este capítulo não tem a pretensão de ser um estudo detalhado da criptografia. O capítulo apresentará, além do conceito de criptografia, seus objetivos, suas técnicas de aplicação, como a substituição, a transposição e a esteganografia, e os modelos criptográficos assimétrico, simétrico e híbrido [4].

No mundo contemporâneo, há muitas motivações para se investir em segurança, principalmente na comunicação em redes de computadores, pois sua popularização dentro da sociedade trouxe uma enorme expansão sobre os negócios que funcionam por ela [4, 11, 13].

A Internet é um bom exemplo deste panorama. Ela é utilizada para os mais diversos fins, como comércio eletrônico, transações financeiras e movimentações bancárias. Todas essas novas utilizações trouxeram inúmeras oportunidades de novos negócios. Porém, tanta facilidade acabou criando, também, novos tipos de crimes e contravenções - os chamados crimes digitais - crimes praticados por meio das redes de computadores [13].

Como uma contra medida a essas infrações, uma série de técnicas de proteção começaram a ser utilizadas. Entre elas, a disciplina da ciência de criptologia utilizada para codificar arquivos de maneira que somente quem possua uma chave de entrada poderá decodificar o mesmo - a criptografia [4].

A necessidade de tornar uma mensagem incompreensível não é algo novo, mas seu conceito foi aplicado às novas necessidades da tecnologia. O objetivo é transmitir as informações de um determinado remetente de uma maneira que somente o destinatário possa compreender, obtendo-se segurança das transmissões de dados [4].

2.1.1. Conceito de Criptografia

A Criptografia é o estudo dos princípios e das técnicas pelas quais a informação pode ser transformada em uma mensagem incompreensível, o que a torna difícil de ser lida por alguém não autorizado, sendo, no entanto, possível que o receptor da mensagem a possa desfazer a transformação e lê-la com facilidade [4, 11].

De fato, a criptografia é mais do que uma técnica de codificação. Na prática, é um ramo especializado da teoria da informação com muitas contribuições de outros campos, como o da matemática [4, 11].

O processo de criptografia, de maneira simplificada, representa um processo, o qual ao enviar uma mensagem, aplica-se sobre ela um tipo determinado de codificação, previamente combinado com o destinatário, por meio de uma chave criptográfica. Ao transmití-la por um meio de comunicação, o destinatário a receberá e usará um método determinado para decodificá-la (com a chave criptográfica). Ao realizar este, o receptor terá acesso à mensagem original [4, 11].

Para o bom entendimento da criptografia, é importante conhecer alguns conceitos. Entende-se por mensagem original qualquer informação que se deseja compartilhar. Dessa forma, a chave criptográfica é o padrão de alteração que a mensagem original será submetida para que resulte na mensagem codificada, ou criptografada, que será transmitida, e o destinatário deve possuir uma chave, que pode ser a mesma ou não, que seja capaz de reverter, ou decifrar, a mensagem ao estado original. Entende-se como codificação de mensagem a aplicação da chave criptográfica à mensagem

original, resultando na mensagem codificada ou criptografada. E a decodificação de mensagem como sendo a aplicação da chave criptográfica à mensagem codificada, resultando na mensagem original [4].

A figura 1 a seguir ilustra o processo de codificação de uma mensagem. Caso a mensagem criptografada seja interceptada na sua transmissão, ela somente poderá ser decifrada se o intruso tiver uma chave criptográfica utilizada no processo. Caso isso não aconteça, ele terá em mãos uma mensagem sem nenhum sentido ou de falsa compreensão [4, 11].

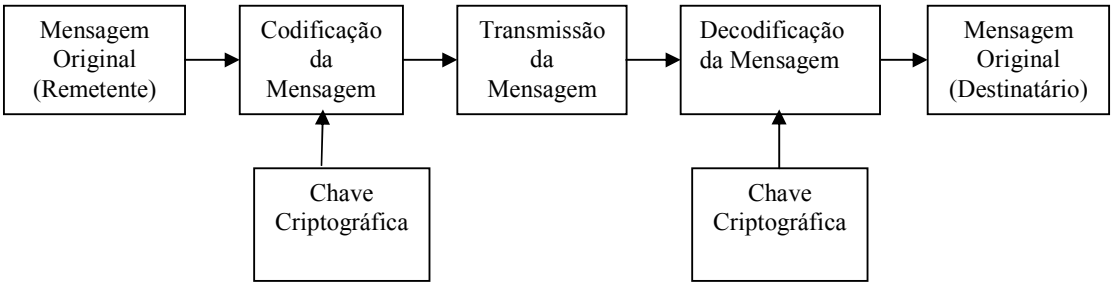

Figura 1 - Sistema de Criptografia

2.1.2. Objetivo da Criptografia e a Segurança da Informação

O objetivo da criptografia é garantir que a informação seja compreensível apenas para quem ela é destinada, e caso ela seja interceptada, o intruso não seja capaz de compreendê-la. Porém, para que a criptografia seja eficaz é preciso pensar também na segurança da infromação.

Quando se pensa em segurança da informação, existem quatro objetivos principais, que por meio da utilização da criptografia, tenta-se garantir para poder atingir o nível desejado de segurança. Esse nível, em geral, é analisado de acordo com o negócio em questão, sempre buscando a melhor relação custo-benefício. São eles [4, 13]:

- Confidencialidade da mensagem: só o destinatário autorizado deve ser capaz de extrair o conteúdo da mensagem da sua forma cifrada. Além disso, a obtenção de informação sobre o conteúdo da mensagem, mesmo com uma distribuição estatística de certos caracteres, não deve ser possível, uma vez que, se o for, torna mais fácil à análise criptográfica;

- Integridade da mensagem: o destinatário deverá ser capaz de determinar se a mensagem foi alterada durante a transmissão; Deve-se gerar um rastro da mensagem original e enviado junto da mensagem, assim o destinatário poderia gerá-lo novamente e compará-lo com o recebido, e tendo que a igualdade de ambos certificaria a integridade da mensagem.

- Autenticação do remetente: o destinatário deverá ser capaz de identificar o remetente e verificar que foi mesmo ele quem enviou a mensagem. O ato de o emissor comprovar sua identidade para uma pessoa, computador ou empresa, utilizando, por exemplo, a Certificação Digital[1];

- Não-repúdio, ou não recusa, do remetente: é a garantia que o emissor de uma mensagem não poderá posteriormente negar sua autoria, visto que somente aquela chave privada poderia ter gerado, por exemplo, aquela assinatura digital. Deste modo, a menos que se comprove o uso indevido do certificado digital, ele não poderá negar a autoria da transação.

[1] É a atividade de reconhecimento em meio eletrônico, que se caracteriza pelo estabelecimento de uma relação única, exclusiva e intransferível entre uma chave de criptografia, seu emissor e a Entidade Certificadora, permitindo a efetivação da autenticação do emissor.

Nem todos os sistemas ou algoritmos criptográficos atingem todos os objetivos citados acima. Sistemas criptográficos mal concebidos ou implementados os atingem só por acidente – “falta de interesse por parte da oposição”. Até sistemas bem concebidos e implementados podem ser, e freqüentemente são, reduzidos pelos usuários ao equivalente a “queijo suíço”, ou seja, cheios de buracos e falhas. Mas, mesmo em sistemas criptográficos bem concebidos, bem implementados e usados adequadamente, alguns dos objetivos acima não são atingidos em algumas circunstâncias. Por exemplo, o sistema pode destinar-se a um ambiente com recursos computacionais limitados, ou pode não interessar a confidencialidade [4, 11, 13].

2.1.3. Técnicas de Criptografia

Existem várias técnicas criptográficas. Elas podem ser lógicas ou não. Com o avanço matemático dentro da computação, surgiu a possibilidade de se utilizar a criptografia com o poder de processamento dos computadores. E, com isso, trabalhar com chaves de alta complexidade. Para isso, é necessário definir regras que possam ser implementadas em linguagem de computação. Entre elas, os métodos lógicos são as técnicas criptográficas de embaralhar a mensagem utilizando uma chave. É importante entender as técnicas atuais para futura comparação com a Criptografia Quântica. Elas são divididas em três classes: Substituição, Transposição e Esteganografia. A seguir todas serão discutidas e exemplificadas individualmente [4].

2.1.3.1. Substituição

A técnica de substituição é de simples funcionamento. Consiste em trocar uma letra ou um conjunto de letras por outra(s) letra(s), símbolo(s) ou

número(s). Porém, esta pode ser quebrada utilizando força-bruta[2] junto com uma análise de freqüência de letras e sílabas de um determinado idioma, facilitando a criptoanálise[3]. A substituição é dividida em dois procedimentos [4, 13]:

a) Monosilábica/Substituição Simples: na qual se substitui cada um dos caracteres do texto original por outro, de acordo com uma tabela definida previamente. Portanto, o comprimento da mensagem cifrada é o mesmo que o da mensagem original [4].

O exemplo a seguir mostra uma tabela de substituição, que no caso faz o papel da chave criptográfica. Por exemplo, no texto o caractere “A” será trocado pelo caractere “H” e assim por diante.

Original	Substituto	Original	Substituto	Original	Substituto	Original	Substituto
A	H	H	O	O	V	V	C
B	I	I	P	P	X	X	D
C	J	J	Q	Q	Y	Y	E
D	K	K	R	R	W	W	F
E	L	L	S	S	Z	Z	G
F	M	M	T	T	A		
G	N	N	U	U	B		

Tabela 1 - Exemplo de tabela para substituição simples definida previamente

Mensagem original:

“MENSAGEM DE EXEMPLO A SER CODIFICADA”.

Mensagem criptografada:

“TLUZHNLT KL LDLTXSV H ZLW JVKPMPJHKH”.

[2] Técnica que utiliza a tentativa e erro, testando todas as possibilidades, para decifrar uma mensagem.
[3] Ramo da criptografia que estuda formas de decodificar uma mensagem sem conhecer a chave.

b) Polialfabéticas – que consistem em substituir um conjunto de letras por outro.

O simples fato de alterar a ordem na seqüência das letras já caracteriza um “novo alfabeto”. Por exemplo, “*z-y-x-w-v-u”*, é um alfabeto de substituição; “*b-a-d-c-f-e”* é um outro alfabeto de substituição diferente. Se ambos forem utilizados para cifrar uma mesma mensagem, substituindo as letras originais, então se trata de uma substituição polialfabética [4, 13].

A substituição polialfabética pode ser com palavra-chave ou com autochave. Num sistema de palavra-chave é essa que indica os alfabetos cifrantes que devem ser usados. Já com autochave, há uma chave que indica a escolha inicial do alfabeto cifrante, e depois a própria mensagem determina os alfabetos subseqüentes [4, 13].

2.1.3.2. Transposição

A transposição, ou cifra de permutação, consiste em embaralhar as letras das mensagens, de acordo com um padrão definido, sem alterá-las. Reorganiza-se, assim, a ordem dos bits, caracteres ou bloco de caracteres. Porém, também não resiste à análise de freqüência das letras do idioma, uma vez que todas as letras da mensagem são mantidas, apenas embaralhadas. A figura abaixo segue como um exemplo [4, 11]:

Figura 2 - Sistema de Criptografia

2.1.3.3. Esteganografia

Define-se esteganografia como sendo a arte e a ciência da comunicação que esconde a existência da comunicação de forma a não permitir que o inimigo, até mesmo, detecte que existe uma mensagem secreta presente [4].

Seu objetivo é esconder mensagens dentro de outras mensagens, e, desta forma, uma mensagem com o método de esteganografia reduz a chance de ser detectada.

Para isso, pode-se utilizar de diversos métodos, por exemplo [4]:

- Tinta invisível;
- Pontos microscópicos;
- Etc.

O uso das tintas invisíveis e pontos microscópicos eram bastante comuns na 2ª Guerra Mundial, principalmente pela facilidade de se encontrar as tintas, como leite e vinagre, e os pontos microscópicos, ou micropontos, para fazer com que suas mensagens viajassem discretamente. Dessa forma, as fotografias eram do tamanho de um ponto e seriam ampliadas posteriormente. Era uma espécie de microfilme colocado numa letra, num timbre, etc. [4].

2.1.4. Modelos Criptográficos

A classificação dos modelos criptográficos baseia-se na forma como a chave será utilizada pelos algoritmos do modelo. Ela pode ser única e secreta - modelo simétrico - ou utilizar um conjunto de chaves pública e privada - modelo assimétrico, ou ainda uma mescla dos dois anteriores – modelo híbrido [4].

2.1.4.1. Simétrico

Os algoritmos que utilizam chave única são denominados de chave simétrica ou privada, e, caracterizam-se por utilizar a mesma chave tanto para a codificação como para a decodificação dos dados [4].

Toda a segurança concentra-se no segredo da chave. Quanto maior ela for, maior a dificuldade de se descobrir a chave. Entretanto, com isso não se garante total segurança da mensagem, pois ao saber o significado da chave, um estranho poderia lê-la, alterá-la e reenviá-la sem que o destinatário e o emissor saibam que o conteúdo foi lido e/ou alterado [4].

O modelo simétrico é bastante utilizado em meios que possuem somente um emissor e um receptor, onde se consiga um canal de comunicação extremamente seguro, imune a terceiros - neste meio que será transmitida a chave criptográfica [4].

Este canal pode ser desde uma conexão direta entre computadores, uma carta, uma correspondência eletrônica ou até uma chamada telefônica entre o emissor e o receptor. O importante é garantir a segurança da chave, peça principal na criptografia simétrica [4].

Pode-se exemplificar a conexão direta entre o emissor e o receptor ao se analisar os terminais bancários, pois são conectados diretamente com a central de dados, utilizando a rede particular do banco em questão. Entretanto, devido a grande quantidade de pessoas que utilizam deste serviço, trocar chaves secretas com segurança ao enviar uma mensagem é praticamente impossível. Porém, é uma situação comum com a Internet. Dessa forma, faz se necessário que cada par emissor-receptor estabeleça a sua própria chave criptográfica. Caso contrário, a chave deixaria de ser secreta, comprometendo assim a segurança da informação [4].

Outro exemplo prático: considere o que teria de ser feito para enviar um memorando confidencial a acionistas de uma empresa. Primeiro, ter-se-ia de entrar em contato com cada acionista, individualmente, para que pudesse fazer a troca da chave secreta. Isso poderia ser feito ao telefone, mas se as mensagens fossem extremamente confidenciais? Talvez fosse melhor você

trocar a chave pessoalmente. Lembre-se de que precisaria fazer isso para todas as pessoas [4].

Cada acionista uma teria uma chave secreta separada. Para aumentar a complexidade desse sistema, também deverá memorizar qual chave serve para cada cliente. Caso as misture, os clientes não serão capazes de ler as suas mensagens, mostrando, assim, que esse tipo de sistema não é viável para transações comerciais comuns devido ao alto número de pessoas envolvidas [4].

O algoritmo mais conhecido de chave única é o DES, *Data Encrypt Standard*, criado no ano de 1977. Ele é uma codificação composta que cifra blocos de 64 bits em outros blocos de 64 bits. Assim o DES faz substituição monoalfabética sobre um conjunto de caracteres. Outro algoritmo bastante popular é o TDES, ou DES3, que nada mais é que a aplicação do DES por três vezes sobre o texto a ser cifrado [4].

No caso do DES, várias tentativas de quebra, criptoanálise, já foram feitas e publicadas. O DES pode ser quebrado pelo método da “força-bruta”, tentando-se todas as combinações possíveis para descobrir a chave, considerado algo não muito difícil para o poder computacional atual [4].

A questão da segurança do DES criou polêmica desde sua criação. Há muitas especulações, inclusive sobre a existência de uma porta oculta, ou *Trap Door*, e também a respeito do número de bits da chave, 56 bits da chave mais 8 bits de paridade totalizando 64 bits, mas ainda não passam de especulações [4].

Em 2001, o *National Institute of Standards and Technology*, NIST, anunciou o novo padrão, denominado AES (*Advanced Encryption Standard*), utilizando o algoritmo *Rijndael*, dos belgas Joan Daemen e Vincent Rijmen. O AES pode processar blocos de 128, 192 e 256 bits com chaves destes mesmos tamanhos. Os possíveis tamanhos de chave aceitos correspondem as diretivas do AES, embora o tamanho oficial seja de 128 bits [4].

A criptografia, por meio da Distribuição de Chave Quântica, a ser explicado adiante, pode ser encaixada em qualquer modelo dependendo do protocolo. O que será utilizado gera uma chave simétrica, porém ela possui

mais algumas características por sua natureza Quântica que irão elevar o nível de segurança [4].

2.1.4.2. Assimétrico

Os algoritmos que utilizam chaves distintas para cifrar e decifrar são também conhecidos como algoritmos de chaves assimétricas e caracterizam-se por utilizar uma chave publica para codificar e outra, a chave privada, para decodificar, e o contrário também é possível. Assim, tanto permitem a proteção da informação, como evitam sua adulteração [4, 11].

Este modelo baseia-se neste conjunto de chaves, uma de domínio público e outra privativa. Por ser pública, uma chave deverá ser divulgada publicamente a todos os receptores, enquanto que a chave privada fica apenas em poder do transmissor. Elas se relacionam, pois a chave secreta é a função inversa da chave pública. A segurança do modelo está baseada na dificuldade da fatoração de números primos grandes, conhecidos como titânicos, explicados adiante [4, 11].

Primeiramente, o par de chaves pode ser utilizado para realizar uma assinatura da mensagem, confirmando a autenticidade da mesma. Para tanto, o transmissor codifica a mensagem utilizando a chave privada e a envia. O receptor pode então usar a chave pública para decodificar a mensagem. O fato de a chave pública conseguir decodificar a mensagem garante a autenticidade da mensagem, como uma assinatura [4].

De fato, a criptografia utilizando a chave privada do transmissor não garante que a mensagem não será lida por outros a não ser o receptor. Uma vez que, a chave que decodifica é a chave pública, que pode estar em poder de qualquer um. Por isso, a criptografia com a chave privada é usada para codificar uma assinatura ao invés da mensagem como um todo [4].

Pode-se, desta forma, criar um *hash*[4] da mensagem e codificá-lo. Mesmo que ocorra a interceptação, ninguém conseguirá adulterar a mensagem, pois não conseguirá produzir um *hash* que seja decifrada pela chave pública, garantindo assim a autenticidade da mensagem. Se o transmissor codificar uma mensagem com a chave pública do transmissor, a mensagem só poderá ser decifrada com a chave privada, garantindo que apenas o receptor lerá seu conteúdo [4].

Exemplificando: considera-se que A e B desejam trocar mensagens entre si, imagina-se que para tanto ambos possuem uma chave pública e uma chave privada. "A" cria uma mensagem e gera um *hash*. Além disso, “A” cifra o *hash* com sua chave privada e em seguida codifica tudo, mensagem e *hash*, com a chave pública de "B" [4].

Apenas "B" poderá decifrar a mensagem usando sua própria chave privada. Então, para confirmar a origem da mensagem, "B" decodifica o *hash* usando a chave pública de "A", gera um *hash* da mensagem e compara o *hash* gerado com o *hash* recebido, determinando que a mensagem tenha sido realmente gerada por "A" [4].

O algoritmo de chave pública mais utilizado atualmente é o RSA[5], que usa o problema intratável da fatoração de inteiros. Multiplicar dois números primos é computacionalmente tratável e muito rápido, porém, a partir deste resultado, descobrir quais os números que multiplicados deram origem a esse, seria computacionalmente intratável. Todavia, nos anos 80, Samuel Yates iniciou uma lista dos “Maiores Primos Conhecidos” e criou o nome Primo Titânico para designar qualquer número primo com 1.000 ou mais dígitos decimais [4, 13].

Os números primos em sua maioria são titânicos e, atualmente, milhares deles são conhecidos. Entretanto, na época em que Yates definiu os primos

---

[4] Hash é uma seqüência de caracteres resultante da aplicação de um método para transformar dados, de tal forma que o resultado seja exclusivo e não possa ser revertido ao formato original. Sua característica principal é a não-duplicidade de dados.

[5] RSA é um algoritmo de codificação de dados baseados em chaves assimétricas, desenvolvido por Ron Rivest, Adi Shamir e Len Adleman.

titânicos, tinha-se conhecimento de poucos. Cerca de dez anos mais tarde, Yates designou como Primo Gigante todo número primo que possuísse 10.000 ou mais dígitos decimais. Nos anos 90, estes primos eram bastante raros. Atualmente, diversos são conhecidos. Há a categoria Megaprimos que são números primos que possuem no mínimo dez mil dígitos decimais [4, 13].

A seguir será mostrada, na tabela, a lista dos dez maiores números primos conhecidos atualmente [4].

Posição	Número	Dígitos	Ano
1	$7068555 \cdot 2^{121301}-1$	36523	2005
2	$2540041185 \cdot 2^{114729}-1$	34547	2003
3	$18912879 \cdot 2^{98395}-1$	29628	2002
4	$1213822389 \cdot 2^{81131}-1$	24432	2002
5	$109433307 \cdot 2^{66452}-1$	20013	2001
6	$984798015 \cdot 2^{66444}-1$	20011	2001
7	$3714089895285 \cdot 2^{60000}-1$	18075	2000
8	$909004827 \cdot 2^{56789}-1$	17105	2005
9	$1162665081 \cdot 2^{55649}-1$	16762	2004
10	$671383317 \cdot 2^{48345}-1$	14563	2004

Tabela 2 - Dez maiores números primos conhecidos

2.1.4.3. Híbrido

Os algoritmos híbridos são a união dos algoritmos anteriores, pois utilizam em conjunto as técnicas simétrica e assimétrica para codificar e decodificar a mensagem. Podem ser utilizados também para a troca de chave criptográfica como, por exemplo, utilizar a técnica assimétrica para transmitir a chave simétrica, ou vice-versa [4, 11].

2.2. Física Quântica

Este capítulo tem por objetivo apresentar os conceitos da Física Quântica que estão relacionados, direta ou indiretamente, com a monografia e o desenvolvimento do protótipo. Além disto, será necessário, para melhor elucidar o objetivo a ser alcançado, descrever alguns conceitos da Física Clássica. É importante ressaltar que este capítulo não tem a intensão de se aprofundar na teoria Física, somente apresentar seus conceitos para o desenvolvimento deste trabalho e da simulação.

Num primeiro momento, será trazido o objetivo, objeto e a definição da Física Quântica. Seus conceitos serão utilizados para delinear uma base teórica na construção do protótipo.

Posteriormente, serão descritos os princípios da Física Quântica e suas principais interpretações que, além de serem analisadas, serão incorporadas a este trabalho com o objetivo de alcançar uma simulação, através do protótipo.

Uma breve análise na história da Física mostra que esta “nasce” com Aristóteles, no século III a.C., e ocupa-se da "substância que tem em si mesma a causa de seu movimento"[6], conforme escreve o filósofo grego em sua obra, Metafísica. Ou seja, a Física Clássica é a teoria do movimento, que é conduzida pela razão e racionalismo na busca da previsibilidade dos fenômenos naturais, no estabelecimento das regras capazes dessa previsão, e que permitam como condição de seu entendimento, a descrição visual do desenvolvimento dos fenômenos, representando sua estrutura por meio de partículas em movimento [5, 6, 13].

---

[6] *Metafísica*, VI, 1, 1025 b 18.[5]

2.2.1. Definição e Objetivo da Física Quântica

O objeto da ciência da Física é o estudo da natureza e seus fenômenos, métodos, fundamentos teóricos e objetivos epistemológicos[7]. Estes estão em relação direta com as concepções que se tem de natureza. Isto equivale a dizer que, do ponto de vista histórico, como acontece com qualquer área do conhecimento, em diferentes épocas existem diferentes modos de conceber a ciência [5, 7, 9].

A Física Quântica tem como objetivo o estudo da natureza em escala microscópica, seus fenômenos e interações, e pode ser definida de diferentes maneiras. Para este trabalho, será apresentada como: a teoria científica que descreve os objetos microscópicos e sua interação com a radiação (luz), ou seja, é a Física dos componentes da matéria – moléculas, átomos e núcleos, que, por sua vez, são compostos por partículas elementares [5, 6, 13].

Uma outra maneira de apresentar a Física Quântica, mais filosófica, seria que o fundamental desta teoria é que o observador não pode ser separado do objeto que está sendo observado. Ou, numa versão mais generalizada, seria que é a teoria que atribui, para qualquer partícula individual, aspectos ondulatórios, e para qualquer forma de radiação, aspectos corpusculares. Essas outras definições, embora estejam corretas, não serão aplicadas diretamente para o contexto deste trabalho [7, 9].

2.2.2. Princípios e Interpretações

Os princípios e as interpretações da Física Quântica ajudam a construir uma base para o desenvolvimento do protótipo. Por isto, será feito uma introdução a eles e, depois, cada princípio e sua utilização neste trabalho serão detalhados.

Na teoria clássica, as equações de movimento, para um determinado sistema, podem ser produzidas de forma a descrever a situação da partícula. Ou seja, sua posição e velocidade, para todos os valores do tempo. Para que isto aconteça, é necessário conhecer as forças que atuam sobre a mesma [7, 13].

Um momento preciso, escolhido como instante inicial da partícula - como se fosse tirada uma foto da mesma - é essencial para analise do seu estado atual e, assim, determinar a forma exata de seu movimento futuro. Esta metodologia foi utilizada com grande sucesso no mundo macroscópico, como por exemplo na astronomia, para prever os movimentos subseqüentes de objetos em função de seus movimentos iniciais [7, 9].

Nota-se, no entanto, que, no processo de realizar análises, o observador interage com o sistema. Mas nesse mundo, a perturbação causada pela interação, ou medição, pode ser desprezada, devido a sua insignificância perante as outras interações em questão [7, 9, 13].

Pode-se extrair desta teoria, como exemplo, a astronomia contemporânea, que possibilita a medição precisa da posição da Lua. A posição deste satélite é perturbada pela medida, mas, devido à sua grande massa, essa excitação pode ser ignorada [5, 7, 9].

Ao analisarmos uma escala menor, como, por exemplo, a de uma experiência microscópica, a interação do observador com o sistema provoca uma perturbação do mesmo, e dado à diferença de massa, esta não pode ser ignorada [5, 7, 9].

Ao se transportarem as análises mostradas acima para a Teoria Quântica, pode-se delinear seus pressupostos básicos. De acordo com esta, os estados das partículas, ou até mesmo as partículas, não são conhecidas até que se faça uma medição. Porém, estes eventos são governados por

---

[7]Refere-se à Epistemologia: estudo crítico dos princípios, hipóteses e resultados das ciências já constituída, e que visa determinar fatos lógicos, o valor e o alcance objetivo delas; teoria da ciência; teoria do conhecimento e metodologia.

probabilidades e ninguém pode afirmar o que irá ou não acontecer. Pode-se somente argumentar em termos de probabilidades [7, 9, 13].

Um dos grandes estudiosos desta área foi o físico alemão Max Planck[8], que afirmava em seus estudos que as coisas mudam quando se olha para elas. Além disso, ele assegurava que a luz era dividida em pacotes de energia, o quanta de energia, denominados posteriormente de fótons [7, 9].

Desta forma, um fóton deve ser completamente absorvido ou refletido, não sendo possível que seja parcialmente refletido e ou parcialmente absorvido. Por outro lado, o conteúdo da onda luminosa pode ser descrito pela interação entre dois campos perpendiculares e variantes no tempo, os campos: elétrico ($\vec{E}$) e magnético ($\vec{B}$). A luz é estudada pelo Princípio da Dualidade Onda-Partícula, que será tratado no próximo tópico, que a conceitua e descreve suas características. Para isto utiliza-se de dois modelos, o ondulatório e o corpuscular [6, 9, 13].

A luz analisada pelo modelo ondulatório possui um plano de polarização que é descrito como o plano que contém o campo elétrico ($\vec{E}$) e a direção de propagação da onda. Com isto, pode-se definir uma atribuição aos diferentes estados de polarização, medidos em graus, de um fóton. Então, se um raio de luz é direcionado a um espelho, como nenhum espelho é perfeito, conclui-se que 19 fótons de 20 são refletidos pelo espelho e o outro é absorvido, supondo uma reflexão de 95%. Mas como saber qual é absorvido e quais são refletidos? Não é possível saber. Um fóton tem 95% de chance de ser refletido e 5% de chance de ser absorvido. Não há nenhuma regra ou propriedade secreta do fóton que possa predizer seu comportamento [13, 14].

[8] Max Planck foi o primeiro físico a propor a hipótese da quantização, em 1900.

2.2.2.1. Princípios da Física Quântica

A Física Quântica possui, como alicerces de sua fundamentação teórica, vários princípios. Porém, este trabalho utilizará três deles [9, 13]:

- Dualidade onda-partícula;
- Princípio da Incerteza;
- Superposição de estados;

Dualidade Onda-Partícula

A dualidade onda-partícula, enunciada por deBroglie em 1924, constitui uma propriedade básica da Física Quântica e consiste na capacidade das partículas subatômicas de terem um comportamento descrito pelos modelos ondulatório e corpuscular de maneira complementar [5, 6, 7, 9, 10, 13, 14].

A ligação dos modelos é realizada por interpretação probabilística. O objeto quântico se divide em duas partes: uma partícula com trajetória bem definida, porém desconhecida, e uma onda associada. A probabilidade da partícula se propagar em uma certa direção depende do quadrado da amplitude da onda associada [5, 6, 7, 9, 10, 13, 14].

Princípio da Incerteza

O princípio da Incerteza, ou também conhecido como princípio de Heisenberg, enuncia que é impossível determinar com precisão arbitrária no mesmo instante a velocidade e a posição de uma partícula. Aplica-se a duas grandezas compatíveis entre si, como nos pares de grandezas posição e momento ou tempo e energia. Isso significa, tomando como exemplo posição x e momento p_x, que uma partícula tem sempre x e p_x definidos, porém desconhecidos. Então ao se medir x, tem-se então uma incerteza grande para p_x, pois a medição em x provocará um distúrbio incontrolável em p_x, visto de

forma geral. Isso será mostrado na simulação, pois ela se baseia nesse princípio para conseguir distinguir o fóton lido do não lido [5, 6, 7, 9, 10, 13, 14].

Superposição de estados

Dados dois estados admissíveis de um sistema quântico, a soma destes dois estados também é um estado admissível do sistema. Ou seja, o estado de polarização de qualquer fóton pode ser representado como uma combinação linear de dois estados ortogonais de polarização. Por exemplo, se um dado fóton percorre um caminho, se for inserido um detector de presença, *a*, supondo 100% de eficiência, ele registrará a presença do fóton. Então se pode atribuir um estado ao fóton ψ_a para o detector *a*. Porém, se utilizar um novo detector de presença, *b*, supondo 100% de eficiência, então pode atribuir um novo estado ao fóton, ψ_b [2, 10, 13, 14].

Esses dois estados têm uma propriedade importante para o trabalho. Se o estado for ψ_a passar por um detector *b*, ou vice-versa, nada será detectado, pois se pode afirmar que são estados ortogonais, são estados que formam um ângulo $\theta = \pi/2$ entre si [2, 10, 13, 14].

Essa polarização, que define o estado a ser considerado, permite fazer uma analogia com o bit, associando uma dada polarização a um valor binário. E essa analogia será denominada de *qubit*, que será explicado no item 2.3.3, mais adiante. A propriedade da ortogonalidade será utilizada pelos protocolos de Criptografia Quântica, como será explicado mais adiante [2, 10, 13, 14].

2.3. Criptografia Quântica

Com a necessidade de se proteger a informação, a complexidade das chaves começou a crescer exponencialmente, sempre na busca de melhorar a segurança. Essa demanda trouxe inúmeros avanços para o campo da criptografia em geral, como os algoritmos simétricos e assimétricos, que são populares e conhecidos livremente. Porém, trouxe junto um outro problema: o aumento da complexidade dos algoritmos criptográficos, que influenciou o crescimento da dificuldade em manusea-lo. E, com isso, aumentou a demanda por poder de processamento para todo o processo de criptografia [7].

Outra grande dificuldade da criptografia é garantir a segurança no momento da troca ou compartilhamento da chave criptográfica nos modelos simétricos. Se ocorrer uma falha de segurança, ou seja, um vazamento da informação (no caso a chave criptográfica), toda a criptografia fica comprometida, pois o detentor da chave poderá interceptar a mensagem, alterá-la ou não, e tanto o emissor quanto receptor não ficarão sabendo da quebra de segurança ocorrida [10, 13].

Para solucionar esse problema foi desenvolvido o modelo assimétrico de criptografia, o RSA, falado anteriormente, porém este tornou a complexidade das chaves ainda maior, pois ele funciona a partir de algoritmos matemáticos baseados na teoria dos números primos [10].

O problema do algoritmo RSA é que ele se baseia na dificuldade em fatorar um número muito grande, como os números titânicos, utilizando um computador clássico, fato que nunca foi comprovado matematicamente; Além disso, existe um algoritmo matemático, o AKS (anexo A) que, em poucas linhas, pode dizer se qualquer número, independente do tamanho, é primo ou não. Isto faz com que os computadores atuais consigam fazer a quebra num tempo bem menor que anteriormente. Entretanto, no dia em que o primeiro computador quântico sair do papel, todos os esquemas públicos de criptografia baseados no RSA serão hipoteticamente invalidados, pois ao realizar uma análise e traçar um paralelo entre eles o estranho notaria que o trabalho para

se quebrar uma chave usando a força bruta seria bem menor e muito mais rápido [8, 10, 13].

A resposta está no conceito de "*one-time pad*", que consiste em utilizar uma chave uma única vez para se codificar uma mensagem e depois descartá-la. Impossibilitando a comparação de duas mensagens codificadas para se chegar a um padrão e descobrir a chave criptográfica. E ele é comprovado matematicamente seguro. Essa chave será gerada de forma aleatória, ou seja, a chave não será criada a partir de nenhuma formula matemática, e assim não existe fórmula matemática que possa quebrá – la [13].

O protótipo, apresentado adiante, simulará como a geração de uma chave baseada na utilização dessa técnica adiciona segurança na utilização do canal de transmissão quântico, para assim garantir a segurança tanto na geração quanto na transmissão, e assim garantindo a segurança da chave.

2.3.1. Conceito de Criptografia Quântica

Dada a necessidade de segurança e as limitações das técnicas de criptografias atuais, viu-se na Física Quântica uma maneira de buscar uma nova técnica que garantisse maior segurança. A Física Quântica descreve observações da natureza, fenômenos naturais. Essas observações, quando definidas e descritas pelo homem, tornam-se leis Físicas [7, 9].

A Criptografia Quântica utiliza um meio de transmissão quântico para garantir a segurança da mensagem, que pode ser a chave criptográfica, baseando-se na natureza Quântica dos fótons. Esses irão ser associados aos bits atuais e assim utilizar as suas propriedades para agregar novas características a criptografia [8, 13, 14].

Atualmente, com o grau de expansão da Internet, a comunicação feita por meio dela é bastante fácil e barata, diferentemente do meio quântico, pouco desenvolvido e conhecido, difícil e extremamente caro.

Para tentar contornar essa dificuldade, foi criado o conceito de *Quantum Key Distribution* (QKD), ou distribuição de chave Quântica. Ele buscaria

transmitir apenas chave criptográfica por causa do nível de segurança oferecido, mas o restante utilizaria meios comuns para transmissão dos dados criptografados para facilitar e diminuir seus custos. É então, fundamental, frisar aqui novamente a idéia da Criptografia Quântica: ela não será utilizada para codificar a mensagem. Até poderia, mas o espião só seria descoberto após a transmissão de parte da mensagem, havendo vazamento de informação, conforme será explicado adiante [10, 13, 14].

2.3.2. Objetivo da Criptografia Quântica

O objetivo da Criptografia Quântica é proteger a mensagem, gerando e transmitindo uma chave criptográfica de forma segura, de maneira que se for detectada por um possível espião, ela se torna inválida e uma nova será gerada. A mensagem torna-se protegida, utilizando-se uma chave válida e segura, e, com isso, impedindo que o espião tenha acesso à mensagem, mesmo que esta seja trafegada em meio de comunicação não seguro [10, 13].

A proteção é feita por meio do fato de que a chave será alterada pelo espião e a taxa de erro do sistema se elevará fazendo com que emissor e o receptor sejam alertados da presença do estranho, invalidando a chave. Assim, o espião não teria acesso à informação, pois a chave que ele possui é inútil e será descartada [10, 13].

Poderia se transmitir à mensagem pelo meio quântico também, porém comparando com a taxa de transferência e o custo das redes públicas, como a Internet, essa se torna inviável [8, 13, 14].

2.3.3. Características da Criptografia Quântica

É das propriedades da radiação luminosa que a Criptografia Quântica obtém suas principais características [7, 9, 13]:

- O Princípio de Heisenberg: garante que qualquer interação com um fóton provoca alteração do mesmo. Assim, se alguém ler a mensagem, esta chegará ao destinatário alterada;

- A dualidade da onda-partícula: o fóton é descrito pelo modelo ondulatório e corpuscular. A ligação dos modelos é realizada por interpretação probabilística da dualidade onda-partícula;

- A superposição de estados: explicam as propriedades de polarização da luz, que estão associadas com suas propriedades corpusculares (atribuídas à polarização aos fótons). Essa polarização que define o estado a ser considerado;

Dessas propriedades Criptografia Quântica herda duas principais características [13, 14]:

- teorema da "*Não-Clonagem*": garante impossibilidade de cópia da informação, pois essa interação provoca a alteração da mesma;

- Capacidade de perceber presença intrusa no meio de comunicação;

Essas duas características são de grande valia para a criptografia, pois incrementam a segurança num padrão nunca antes alcançado [13, 14].

Com isso, surge a necessidade de se ter uma convergência entre fótons e bits. Como se relacionar os bits dos computadores atuais, já que ainda não

dispomos de computadores quânticos, com os fótons que serão transmitidos? *Qubits* são a resposta, associações feitas entre os bits e os fótons. Associa-se o valor de cada bit a um único fóton com um determinado estado de polarização [8, 10, 13, 14].

Quando um fóton viaja pelo espaço ele possui uma vibração. Tomando quatro fótons, como exemplo, todos os quatro fótons viajando na mesma direção, mas o ângulo de vibração de cada um é diferente associado, por exemplo, à vibração do campo elétrico de uma onda eletromagnética clásica. O ângulo de vibração da partícula é chamado de polarização, e uma fonte de luz qualquer, como uma lâmpada incandescente, por exemplo, gera fótons com diversas polarizações. Alguns vibrarão horizontalmente, alguns verticalmente, e outros em todos os ângulos possíveis entre esses dois casos [13, 14].

Colocando um filtro, por exemplo um polaróide, no caminho dos fótons, é possível assegurar que os feixes de luz emergentes consistirão de fótons que vibram em uma direção particular. Em outras palavras, todos os fótons emergentes têm a mesma polarização. Os fótons que estiverem polarizados no mesmo ângulo do filtro passarão livremente. Já os fótons que estiverem polarizados em um ângulo perpendicular ao filtro serão bloqueados e os estados intermediários terão probabilidades de ser transmitidos ou não dependendo do ângulo entre a polarização do fóton e filtro [8, 10, 13, 14].

Essa polarização é associada aos bits utilizando bases de estados ortogonais, e, assim, tornando possível um feixe de fótons polarizados representar uma seqüência binária [8, 10, 13, 14].

Uma vez definido um padrão de polarizações ortogonais, os bits serão representados. Por exemplo, pode–se definir uma base “A” como sendo polarização vertical, ou seja, 90°, como valor um (1) e polarização horizontal como sendo zero (0), ou seja, 0°. Os fótons podem assumir diversas polarizações, porém consegue-se por meio dos filtros determinar que um fóton obtenha uma dada polarização. A polarização a ser utilizada é definida de acordo com a necessidade. Pode-se também ter mais de uma base, como, por exemplo, definir outra base, base B, com polarizações diagonais, de 45° para zero ou um e 135° para um ou zero [10, 14].

2.3.4. Protocolos Quânticos

Há várias maneiras de se definir o que venha a ser um protocolo. Do ponto de vista da computação, poderia se definir como um formato estabelecido para a transmissão de dados entre dois dispositivos computacionais; ou o conjunto de regras sintáticas e semânticas que determinam o comportamento de entidades que interagem, incluindo a seqüência, temporização, formato e controle dos fluxos e erros. Hoje, existem vários protocolos de criptografia, ou seja, diferentes conjuntos de procedimentos, que permitem a troca de chaves entre emissor e receptor de maneira segura e eficiente, que buscam utilizar a Criptografia Quântica como uma técnica para proteger a informação [10, 13].

Estes protocolos definem o tipo de consistência e checagem de erros, o método de compressão de dados, a forma como o dispositivo de envio indicará que a mensagem está terminada e a forma como o dispositivo de recebimento indicará que recebeu a mensagem [10, 13].

Todos os protocolos têm um objetivo em comum: atingir uma das principais metas da criptografia, a distribuição segura de uma chave criptográfica. A seguir serão apresentados os principais protocolos quânticos estudados e desenvolvidos atualmente [10, 13].

2.3.4.1. BB84

Antes de iniciar a explicação deste protocolo, vale a pena ressaltar que ele é usado em todos os sistemas bem-sucedidos de Criptografia Quântica instalados até hoje e, além disso, é o único oferecido por companhias especializadas em segurança de transmissão de dados. Assim, mesmo sendo o primeiro protocolo proposto e inaugurar o estudo e desenvolvimento da disciplina da Criptografia Quântica, ele ainda é, apesar de existirem outras alternativas de Criptografia Quântica apresentadas *a posteriori*, aquele de

maior importância prática e comercial e, por isso, será utilizado no simulador para gerar uma chave criptográfica [2, 10, 13].

O protocolo BB84 foi apresentado em 1984. Escrito por Bennett e Brassard até então é considerado o mais bem elaborado conjunto de procedimentos que permite a troca de chaves entre emissor e receptor de maneira segura e eficiente [2, 10, 13, 14].

Além de tratar a transmissão dos fótons pelo canal quântico, o protocolo BB84 trata o processo de comparação das leituras efetuadas. Ou seja, o emissor e o receptor comparam os resultados obtidos com os enviados. O resultado desse processo é uma chave filtrada que ainda pode conter erros, cujo procedimento de tratamento destes também é abordado pelo protocolo [10, 13, 14].

O BB84 utiliza-se de um sistema quântico de dois níveis. Assim, os estados ou kets[9] |0> e |1> representam fótons linearmente polarizados em direções ortogonais. Por exemplo, os estados |0> e |1> podem representar fótons que se propagam na direção z com campos elétricos oscilando no plano *xy*. As direções de polarização são representadas por vetores unitários [10, 13].

Segundo Rigolin e Rieznik [2, 10], estudiosos da área, Alice e Bob, um receptor e outro emissor, devem primeiramente escolher duas bases que serão utilizadas para a transmissão e recepção dos fótons. Cada base é composta por dois estados ortogonais de polarização. Eles podem escolher, por exemplo, polarizações contidas no plano xy. Tomando ø = 0 ou 0° ø = π/2 ou 90° definimos as direções de polarização de uma das bases (base A). Usando ø = π/4 ou 45° e ø = 3π/4 ou 135° obtemos a outra (base B).

Além disso, eles devem combinar previamente quais estados ortogonais de cada uma das bases representam o bit 0 e o bit 1. Isso pode ser feito via qualquer canal de comunicação. Neste exemplo, utiliza-se os fótons polarizados na direção ø = 0 ou ø = π/4 para representar o bit 0 ($|0\rangle_A$ e $|0\rangle_B$) e

[9] Representação simbólica do bit associado ao estado de polarização, exemplo: |0> ou |1>, conhecido como Ket. Também representados como $|0\rangle_A$ ou $|1\rangle_A$, valores associados para uma base conhecida como A.

aqueles com polarização na direção ø = π/2 ou ø = 3π/4 representando o bit 1 ($|1\rangle_A$ e $|1\rangle_B$). Nesta notação, o sub-índice em cada ket indica fótons polarizados nos auto-estados da base A ou base B [2, 10, 13].

Alice, para transmitir a chave, procede da seguinte forma: primeiro, ela escolhe uma seqüência aleatória de bits para enviar a Bob; depois, qual a base será utilizada para transmitir cada bit. Ela pode transmitir os bits utilizando da Base A ou Base B, escolhendo de forma aleatória. Dessa forma, ela estaria enviando a Bob uma seqüência de fótons representados, por exemplo, pelos seguintes kets: $|0\rangle_A$; $|0\rangle_A$; $|1\rangle_B$; $|1\rangle_B$; $|1\rangle_B$; $|1\rangle_A$; , etc. [10, 13].

Bob, por sua vez, deve escolher apenas qual base ele irá utilizar para detectar cada fóton. Ele oscila entre as bases A e B aleatoriamente. Após a transmissão e detecção dos fótons, Alice e Bob revelam publicamente quais bases utilizaram para enviar e detectar cada fóton, respectivamente. Mas Alice não revela se enviou 0s ou 1s e Bob não revela o resultado de suas medidas. Apenas as bases utilizadas (base A ou base B) são publicamente reveladas [2, 10, 13].

A seguir, eles consideram apenas os resultados nos quais ambos utilizaram a mesma base, descartando todos os demais. Agora, eles revelam publicamente uma parte destes resultados (metade, ou um terço, por exemplo). Se Eva, uma espiã ao sistema, não monitorou a transmissão, os resultados revelados por Bob e Alice devem coincidir; mas se ela o monitorou, a probabilidade de que todos os dados públicos coincidam tendem a zero conforme aumentam os números de bits comunicados, baseado na probabilidade de Eva acertar todas as escolhas feitas de maneira aleatória [10, 13].

Se os dados revelados publicamente coincidirem, isso será uma forte evidência, praticamente uma prova, de que Eva não monitorou a transmissão e eles podem usar o restante dos dados como a chave [10, 13].

Para simplificar a demonstração e sem perda de generalidade, suponha que Alice, Bob e Eva utilizem metade das vezes a Base A e metade das vezes a Base B, Alice para transmitir e Eva e Bob para detectar os fótons. Se Alice e Bob utilizam a mesma base, a probabilidade de Eva usar a mesma base vale 0,5. Agora, se Eva utiliza para monitorar os fótons a outra base, a probabilidade

de Bob medir corretamente o valor do bit transmitido é de apenas 0,5 e não 1, como deveria ser se não tivéssemos um espião ou se Eva tivesse optado pela base correta [10, 13].

O fato de Eva escolher a base errada implica, para um evento, uma probabilidade igual a 0,5 de Bob detectar o valor correto para o bit transmitido por Alice. Para uma chave muito grande, a probabilidade de Bob detectar todos os bits corretamente, com Eva interferindo, tende a zero ou, mais rigorosamente, a $(0,5)^n$, onde n é o número de vezes que Eva usou a base errada. Vale a pena lembrar que estados quânticos arbitrários não podem ser clonados. Isso garante que Eva não pode simplesmente duplicar o estado quântico dos fótons enviados por Alice, medir um deles e enviar a Bob o outro [10, 13].

2.3.4.2. B92

Através do desenvolvimento do protocolo B92, também conhecido como *Two-State*, Bennett anunciou que dois estados eram suficientes para a Criptografia Quântica. Somente dois estados não-ortogonais seriam suficientes. A segurança da Criptografia Quântica confia na inabilidade de um adversário para distinguir uma não ambígua e nenhuma perturbação entre os diferentes estados que o Emissor pode enviar ao Receptor. Consequentemente, dois estados são necessários e, se eles são incompatíveis, ou seja, não mutuamente ortogonais, então dois estados também são suficientes [3, 10, 13].

Para começar a distribuição de chaves implementando este protocolo, Alice e Bob devem combinar inicialmente quais estados |A> e |B> serão utilizados e qual corresponderá ao bit 0 e ao bit 1. O fóton com estado Ø = 0 para |A> e Ø = π/4 para |B>. Alice então transmitirá para Bob uma seqüência de bits codificados, usando os estados |A> para bit 0 e |B> para bit 1. Bob, por sua vez, escolhe aleatoriamente para cada estado recebido de Alice qual medida realizará: P_a ou P_b, onde P_a e P_b são operadores de projeção em espaços ortogonais a |B> e |A>, respectivamente [3, 10, 13].

Dessa forma P_a anula |B> e P_b anula |A> (P_a|B> = 0 e P_b|A> = 0). Terminada a transmissão, Bob anuncia publicamente para quais bits ele obteve resultados positivos sem, no entanto, informar o tipo de medida feita (se P_a ou P_b). São estes bits que serão utilizados por Alice e Bob para obter a chave criptográfica. Como nos outros protocolos, alguns destes bits devem ser sacrificados para checar se Eva monitorou a comunicação. Assim, Bob publicamente informa que base utilizou para medir alguns de seus fótons. Se Eva não interferiu, todas as medidas nas quais Bob obteve um resultado positivo devem corresponder a duas únicas possíveis situações: Alice enviou um estado |A> e Bob mediu P_a ou Alice enviou |B> e Bob mediu P_b [3, 10, 13].

Caso ocorra um evento positivo para uma outra situação, Alice e Bob descartam seus bits, pois Eva interferiu na transmissão. Se apenas estes dois eventos positivos ocorreram, eles têm certeza da segurança da chave, a qual é constituída pelos bits restantes [3, 10, 13].

Após várias demonstrações experimentais simples, constatou-se que, na prática, a solução não é eficiente. Ambos os protocolos, BB84 e B92, são complicados de serem implementados, porém o BB84 é considerado mais seguro do que B92. A diferença está no fato em que o B92 está baseado em apenas dois estados, que não podem ser ortogonais, e o BB84 é baseado em 4 estados [3, 10, 13].

2.3.4.3. SIX-STATE

O protocolo Six-State, diferente do que foi proposto nos protocolos anteriores, onde foi mostrado que dois estados bastam e quatro é o padrão, usa um protocolo de seis estados [13].

Os seis estados constituem três bases. Conseqüentemente, a probabilidade que o Emissor e o Receptor escolham a mesma base é de somente 33%, mas a simetria deste protocolo simplifica significativamente a análise de segurança. Além disso, reduz muito as informações que o espião

obtém para determinadas taxas de erro, típica de um fenômeno de propagação [13].

Cada base é composta por dois estados ortogonais de polarização. Eles podem escolher, por exemplo, polarizações contidas no plano *xy*, assim definindo as três bases que serão utilizadas na polarização dos fótons [13].

2.3.5. Vantagens e Desvantagens

Ao se comparar os anos de estudo da Física com a Física Quântica, notamos o quão recente é esta última. A teoria Quântica é base para os grandes avanços que se apresentam temos hoje em termos de tecnologia, como a nanotecnologia presente nos processadores dos computadores atuais. Ela acrescenta ao protótipo idealizado neste projeto características delineadas nas leis da Física, e quebrá-las seria desafiá-las. Por isto que é incondicionalmente segura e agrega uma grande vantagem sobre as demais técnicas de criptografias [13, 14, 15].

Apesar de ainda estar em pleno desenvolvimento, a sua aplicação tem se mostrado muito eficaz, possibilitando melhorias em várias áreas de pesquisas científicas. Ela não possui todo o seu potencial desenvolvido e explorado. Este fato é uma grande vantagem, visto que ainda pode contribuir com novas descobertas e trazer novos benefícios que virão a agregar maior valor a ela [13, 14, 15].

O estudo dos princípios da teoria Quântica é de grande importância para o desenvolvimento do protótipo. Esses trazem consigo características únicas, que serão utilizados de maneira a acrescentar novos benefícios, antes inexistentes [13, 14, 15].

Por exemplo, nesse trabalho sobre a Criptografia Quântica, deseja-se saber se existe alguém escutando o meio de transmissão, então se transmite uma mensagem, sem importância, e compara-se os resultados obtidos. Caso o espião realmente exista, a sua interferência no momento em que ele lê a mensagem provocará a alteração da mesma, e na comparação posterior da

mensagem serão observadas as diferenças, que serão causadas pelo espião. E, assim, sabendo-se que o meio é seguro, poderia executar-se a troca de chave criptográfica de forma segura [13, 14, 15].

Um aspecto peculiar da Física Quântica, que pode gerar dificuldade no seu estudo, é relacionado à imprecisão da utilização de conceitos clássicos como partículas e ondas para explicar o comportamento dos fótons, pois eles não se caracterizam como partículas, nem tampouco como ondas da Física clássica e sim como em um ente que ora apresenta comportamento ondulatório e ora comportamento corpuscular, sendo as descrições complementares [13, 14, 15].

Outro ponto negativo é que, para se construir um meio quântico, é necessário conseguir um isolamento perfeito, sem interferências externas, porque qualquer alteração do campo eletromagnético, como um fóton aleatório, pode alterar o estado dos *qubits* [13, 14, 15].

Essas desvantagens fazem com que aumente a dificuldade em se trabalhar com os fótons e acabam funcionando como limitadoras na sua utilização, como por exemplo a distância máxima de transmissão de dados bastante reduzida. Porém, tecnologias que estão em pleno desenvolvimento para diminuir esses fatores limitantes tendem a melhorar os polarizadores, as fibras ópticas e outros [13, 14, 15].

3. Métodos / Metodologia

Este capítulo irá descrever todo o processo da simulação, tanto para a transmissão Quântica, quanto para a geração de uma chave criptográfica Quântica, utilizando um protocolo quântico para isso. Além disso, este capítulo irá abordar a base de desenvolvimento do simulador, a linguagem utilizada, e as escolhas e decisões tomadas para a simulação.

3.1. Tecnologias

Antes de abordar o desenvolvimento do simulador, é importante ressaltar as tecnologias que foram utilizadas como base para a sua construção. A seguir elas serão relacionadas de acordo com a relevância para o projeto:

- Servidor de aplicação Web Apache Tomcat;
- Linguagem de programação Java, JavaScript e HTML;
- Ferramenta de desenvolvimento Eclipse;
- Navegador ou *Browser.*

O desenvolvimento do simulador foi baseado em servidor de aplicação Web, para isso foi escolhido um servidor gratuito, que utiliza a linguagem JAVA e bastante popular por sua facilidade e versatilidade, o Apache Tomcat. (http://tomcat.apache.org)

O simulador poderia ser desenvolvido em qualquer linguagem de programação existente hoje. A linguagem Java[10] possui alguns diferenciais em relação as demais, como a sua portabilidade[11], grande aceitação dentro do meio acadêmico, possui documentação vasta através de suas *API's*

[10] Para maiores informações, viste o site http://java.sun.com.
[11] As classes Java compiladas não necessitam de recompilação caso haja mudança no sistema operacional.

(*Application Programming Interface*)[12] para programação Web, conhecidas como *JSP's*, *Java Server Pages* e *servlet's*, nome das classes que processam as requisições ou pedidos dos navegadores. Ainda são utilizadas outras duas linguagens, o JavaScript e o HTML. O primeiro, apesar das semelhanças com o Java ficarem apenas no nome, é uma linguagem de programação criada pela Netscape, em 1995, para ações dinâmicas dentro do próprio navegador. Já o HTML, sigla para *HyperText Markup Language*, é uma linguagem de marcação utilizada para produzir páginas na Internet, pois os códigos HTML são interpretados pelos navegadores.

A ferramenta de desenvolvimento, também conhecida como *IDE*, interface de desenvolvimento, foi o Eclipse, ferramenta gratuita para desenvolvimento em Java que possui inúmeros recursos que aumentam a produtividade e diminuem o tempo de desenvolvimento. Para maiores informações: http://www.eclipse.org.

Para visualização/execução da simulação é necessário um navegador/*browser* instalado na máquina cliente, pode ser o que vem junto ao sistema operacional. Nesse caso será usado o navegador Internet Explorer, da Microsoft. (http://www.microsoft.com)

O simulador foi construído utilizando um padrão de aplicação cliente/servidor. O servidor, neste caso o *Tomcat*, recebe as requisições do cliente, neste caso o navegador escolhido. Essas são enviadas para a *servlet* de destino, especificada na página exibida. Ou seja, as requisições são as entradas e saídas do simulador, todas as informações que foram selecionadas na página serão enviadas para sua controladora específica, que as receberá e, através das informações recebidas, conhecerá o procedimento a ser executado e a página para qual a requisição será respondida.

As informações trocadas entre as *servlets*, as classes controladoras, e as páginas, as *JSP´S*, são armazenadas em vetores unitários, um para cada participante da simulação em questão. Esses vetores armazenam toda

[12] API é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades.

informação relevante, como a seqüência de bits, as conversões, as bases utilizadas e etc.

Todas as escolhas ditas aleatórias, inclusive a perda de fótons, foram feitas utilizando a classe "*RANDON*" da *API* Java. Essa possui métodos que quando chamados devolvem números inteiros ou não, essa decisão é feita através de qual método será chamado, e o simulador não tem controle nenhum sobre os números gerados. Essa classe é utilizada para a geração da seqüência de bits a ser transmitida, para a escolha da base de conversão nos casos onde existem mais de uma base, e também para as escolhas regidas pela probabilidade de acerto ou erro na leitura. Todas as informações trocadas são exibidas nas *JSP'S*, interpretadas pelo navegador que as exibem ao usuário do simulador. Para maiores esclarecimentos sobre o código do simulador, este está em anexo a este trabalho (anexo B).

3.2. Desenvolvimento

Definidas as tecnologias que serão utilizadas na construção e aplicação do simulador, serão definidas duas funcionalidades dele agora:

- Simulação da transmissão de dados em um canal quântico;
- Geração de uma chave criptográfica utilizando um protocolo quântico, o BB84;

A seguir serão mostradas as duas funcionalidades, primeiramente a transmissão e posteriormente a geração de uma chave Quântica. Importante ressaltar que todas as funcionalidades a seguir foram desenvolvidas no simulador, aqui apresentadas com apenas alguns valores reduzidos por conveniência de explanação.

A simulação da transmissão de dados, em um canal quântico, será feita transmitindo uma seqüência de bits definidos pelo emissor, Alice. Para que

essa transmissão seja possível, é necessário antes que seja definida uma base comum de polarização de fótons entre o emissor e o receptor, no caso Alice e Bob. Essa base pode ser definida via qualquer canal que eles decidirem como a Internet, por exemplo.

A base será definida da seguinte forma:

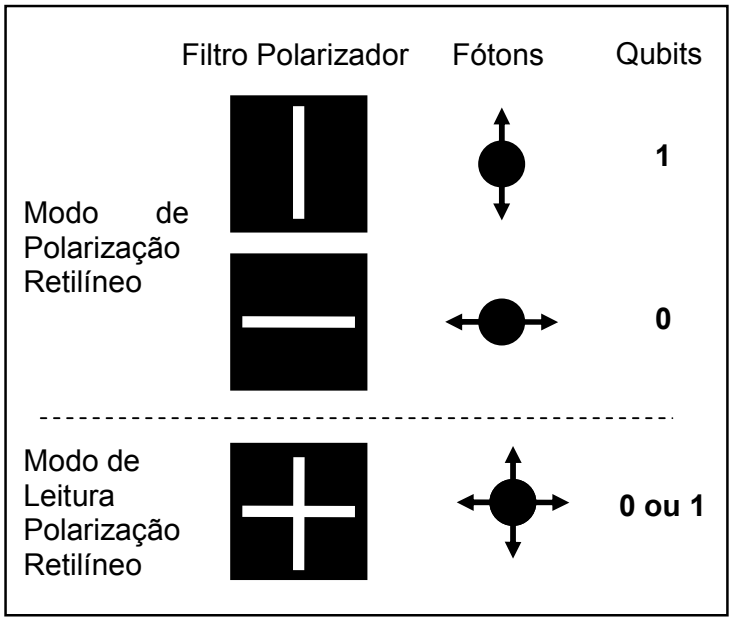

Figura 3 - Base de polarização para a Simulação da Transmissão em meio quântico

Definida a base, Alice irá enviar uma seqüência de fótons que serão polarizados de acordo com a seqüência de bits, que ela deseja transmitir.

Por exemplo, Alice deseja transmitir a seqüência *00101011* para Bob, ela então irá polarizar cada fóton da seguinte maneira:

0	0	1	0	1	0	1	1
							

Tabela 3 - Polarização utilizada em acordo com os bits enviados.

Na tabela anterior, a primeira linha é a seqüência de bits que Alice, o emissor, deseja transmitir e a segunda linha é o filtro polarizador que será aplicado a cada fóton, para se transmitir o fóton de acordo com a base.

No simulador desenvolvido, a seqüência de bits será definida pelo usuário que estará simulando, podendo ser escolhido entre alguns valores possíveis. Esses valores foram definidos tentando sempre a melhor exemplificação, mas eles poderiam ser quaisquer números inteiro.

Bob, por sua vez irá fazer a leitura utilizando a base pré-definida, da seguinte maneira:

							
0	0	1	0	1	0	1	1

Tabela 4 - Seqüência de filtros utilizados para a leitura.

Na tabela anterior, a primeira linha é o filtro aplicado por Bob sobre o fóton para realizar a leitura da sua polarização e a segunda linha é o resultado obtido da leitura, ou seja, o valor em bits que foi associado ao estado de polarização lido, segundo a base definida.

Transmissão feita, como saber se o meio é seguro e se não existe um intruso no meio? Para poder identificar se existe a presença de um espião, Eva, Alice e Bob devem trocar os resultados obtidos, nesse caso a informação trafegada não tem importância, pois foi um teste para o canal, porém caso haja a presença do espião, a comparação apresentará divergências, e assim será identifica se a presença de Eva, independente se a tentativa foi de leitura ou alteração.

A tabela a seguir irá ilustrar todos os passos da transmissão, com as seguintes considerações:

1. O meio é considerado ideal, ou seja, não existe perda de fótons para o meio ambiente ou por impurezas do canal;[13]
2. Eva tentará ler todos os fótons com filtros desconhecidos por Alice e Bob.

Bits que Alice deseja enviar	0	0	1	0	1	0	1	1
Fótons enviados por Alice	|0>	|0>	|1>	|0>	|1>	|0>	|1>	|1>
Bits recebidos por Bob	0	1	0	0	1	1		1
Compartilhamento entre Alice e Bob	**C**	**E**	**E**	**C**	**C**	**E**	**P**	**C**

Tabela 5 - Transmissão de dados pelo canal quântico entre Alice e Bob, com a intervenção de Eva.

A tabela acima exemplifica como funciona a intervenção de Eva na comunicação entre Alice e Bob. Analisando as medições compartilhadas, mostra que Eva utilizou-se de mais de um tipo de filtro, um deles acertou a medição e não adicionou erro no canal, um ou mais filtros resultaram valores na medição e introduziram erro ao canal e um terceiro filtro bloqueou a passagem do fóton, ficando clara a presença de um intruso no canal.

Agora, será mostrada a geração e distribuição da chave Quântica, seguindo o protocolo quântico BB84. Assim como na transmissão, no protocolo existe a necessidade de se definir bases, no BB84 são necessárias duas bases e algumas considerações. No exemplo utilizado, Alice e Bob irão definir as bases que serão utilizadas na distribuição das chaves Quânticas. Essas bases podem ser combinadas via canal publico de comunicação, pois mesmo que ela seja do conhecimento do espião, Eva, ainda sim é possível detectar sua presença.

Como considerações, serão feitas algumas escolhas para que a simulação seja objetiva:

[13] Em caso de se considerar um meio real, definir um nível de perda de fótons aceitável.

1. O meio é considerado na simulação, pode ser ideal ou real, com possibilidade de perda. Ou seja, pode ou não existir perda de fótons para o meio ambiente ou por impurezas do canal;

2. Será dividida em dois momentos distintos, com e sem a participação do intruso, assim posteriormente pode-se comparar os dois resultados;

Na simulação será definido também o tamanho da seqüência de bits que será transmitida. Isso não é definido no protocolo, pois depende do nível de segurança desejado, quanto maior a seqüência maior será a sensibilidade de detecção do intruso e maior a chave gerada. No simulador, serão oferecidas algumas opções, mas nada impede que se defina outros valores.

As duas bases que serão utilizadas na simulação são definidas a seguir:

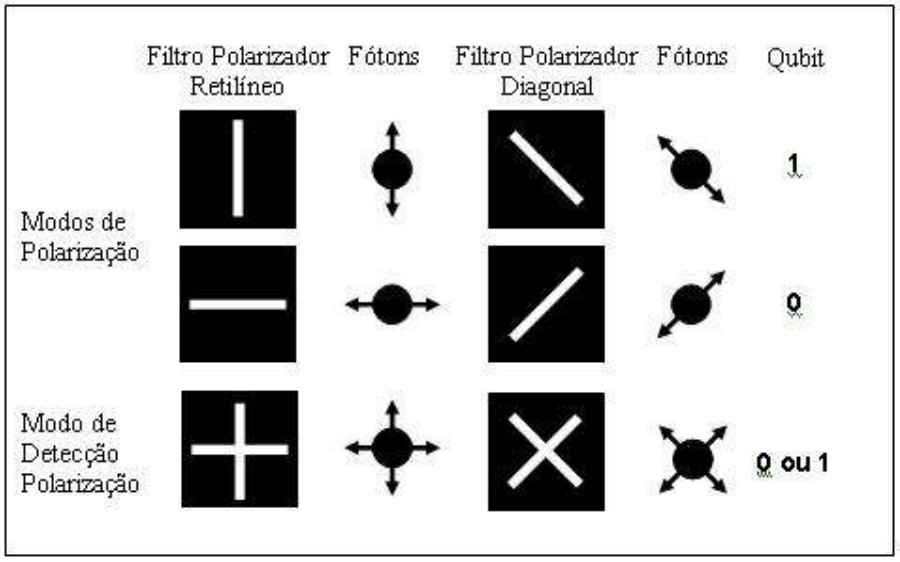

Figura 4 - Filtros de Polarização utilizados na *QKD*

Definidas as bases, Alice irá gerar uma seqüência de bits de forma aleatória, ou seja, não será utilizado nenhum método matemático. De parte desta seqüência de bits que resultará a chave criptográfica.

A seguir a tabela ilustrará as ações de Alice:

Seqüência de Bits aleatória gerada por Alice	1	1	0	1	0	0	0	1
Bases escolhidas de forma aleatória por Alice para a polarização dos Fótons	|A>	|B>	|B>	|A>	|A>	|B>	|B>	|A>
Fótons transmitidos	$\|1\rangle_A$	$\|1\rangle_B$	$\|0\rangle_B$	$\|1\rangle_A$	$\|0\rangle_A$	$\|0\rangle_B$	$\|0\rangle_B$	$\|1\rangle_A$

Tabela 6 - Valores escolhidos ao acaso por Alice para enviar a Bob

Neste momento existem escolhas de caminhos a se seguir: a geração da chave criptográfica com e sem a presença do intruso, no caso Eva, e se será considerada a interferência do meio. Isto se faz necessário, pois os valores apurados são diretamente influenciados por essas decisões. Como o objetivo é a simulação da geração de uma chave quântica, a interferência do meio será mostrada apenas no simulador, pois a seguir será apresentado apenas um exemplo que ajude no entendimento. Então, no primeiro momento, a geração será mostrada sem a presença do intruso e em seguida com a presença do intruso, porém partindo da mesma transmissão da Alice, e assim podendo comparar os dois resultados por terem o mesmo ponto de partida e só a influencia de Eva que irá alterar os resultados encontrados. As escolhas de Bob, a seqüência dos filtros de leitura, também serão mantidas iguais.

Terminada a transmissão de Alice, Bob irá agora escolher uma seqüência aleatória de filtros para ler os fótons recebidos. Na tabela a seguir serão mostrados os filtros e os valores lidos por Bob, sem a intrusão de Eva.

Bases escolhidas de forma aleatória por Bob para a leitura dos Fótons	|A>	|A>	|B>	|B>	|A>	|B>	|B>	|B>
Valores resultantes das leituras	1	1	0	0	0	0	0	0

Tabela 7- Bases escolhidas por Bob para a leitura dos fótons enviados por Alice

Terminado todo o processo de transmissão e leitura, agora Alice e Bob trocam informações sobre as seqüências de bases utilizadas, dela para enviar e dele para ler, como mostrado na tabela a seguir:

Bases escolhidas de forma aleatória por Alice para a polarização dos Fótons	|A>	|B>	|B>	|A>	|A>	|B>	|B>	|A>
Bases escolhidas de forma aleatória por Bob para a leitura dos Fótons	|A>	|A>	|B>	|B>	|A>	|B>	|B>	|B>
Comparação entre Bases escolhidas por Alice e Bob	**C**	**E**	**C**	**E**	**C**	**C**	**C**	**E**
Valores definidos por Alice	**1**	**X**	**0**	**X**	**0**	**0**	**0**	**X**
Valores definidos por Bob	**1**	**X**	**0**	**X**	**0**	**0**	**0**	**X**

Tabela 8 - Comparação de Bases entre Alice e Bob

Na tabela anterior, foi mostrada a comparação, acrescida de duas novas linhas. A primeira compara as escolha de Alice e Bob, onde “C” significa certo, mesma escolha para ambos, e “E” significa errado, escolhas diferentes entre ambos. E a segunda representando os bits que formarão a chave, onde “X” significa que aquele valor foi descartado, independente de acerto ou erro, pois a base foi errada. Para a chave gerada, são considerados válidos apenas os valores lidos onde houve concordância na escolha da base, ou seja, apenas onde as bases escolhidas foram as mesmas. No exemplo acima, a chave resultante foi *“10000”*. Essa chave gerada ainda não é a chave criptográfica.

Definida a chave, o próximo passo é a verificação se houve ou não interferência. Para isso, necessita-se compartilhar parte da chave gerada anteriormente. Assim, Bob define uma parte da chave a ser enviada para Alice comparar. Bob define *"100"*, seguindo o exemplo, ou seja, os três primeiros valores encontrados, e envia a Alice, por um canal de comunicação publico. Alice recebe a chave de comparação e retorna o resultado para Bob, com na tabela a seguir:

Chave gerada em Alice	1	0	0
Chave gerada em Bob	1	0	0
Comparação de Chaves	**C**	**C**	**C**

Tabela 9 - Comparação entre as chaves compartilhadas.

Efetuada a comparação, Alice responde a Bob que as medições foram todas corretas, então ambos descartam os bits compartilhados e a chave criptográfica será o restante dos bits gerados, no exemplo seria *"00"*. Todas as comparações corretas significam que não houve tentativa de leitura dos fótons além das feitas por Bob.

O próximo passo é a inclusão do intruso no meio quântico de comunicação. Para efeitos de comparação, a transmissão inicial de Alice, será considerada a mesma e as bases utilizadas também. Então, antes de Bob tentar fazer a leitura dos fótons, Eva irá utilizar seus filtros para capturar a informação.

Para isso, Eva precisará de filtros também. Num cenário real, Eva pode ter conhecimento ou não das bases definida por Alice e Bob e, na pior situação, Eva terá. Na simulação será usada a pior situação possível, ou seja, que Eva conheça o sistema de bases utilizado por Alice e Bob, pois eles foram trocados por um canal público e capturados por Eva.

Definido os filtros de Eva, ela terá que fazer escolhas sobre qual o filtro utilizará para ler cada fóton, como na tabela a seguir:

Bases escolhidas por Eva para a leitura dos Fótons	|A>	|A>	|A>	|B>	|B>	|B>	|A>	|B>
Valores resultantes das leituras	1	1	0	0	0	0	0	0

Tabela 10 - Filtros utilizados e valores lidos por Eva.

Então os fótons chegam a Bob, que vai realizar a sua leitura dos mesmos. Para comparação, a seqüência de bases de Bob é a mesma da situação anterior. A seqüência escolhida será conforme a tabela a seguir:

Bases escolhidas de forma aleatória por Bob para a leitura dos Fótons	|A>	|A>	|B>	|B>	|A>	|B>	|B>	|B>
Valores resultantes das leituras	1	0	1	1	0	0	1	0

Tabela 11 - Filtros utilizados e valores lidos por Bob.

Feita todas as leituras, Alice e Bob trocam suas seqüências de bases para saber quais serão os resultados que serão ou não descartados. A tabela mostra o resultado das comparações.

Bases escolhidas de forma aleatória por Alice para a polarização dos Fótons	|A>	|B>	|B>	|A>	|A>	|B>	|B>	|A>
Bases escolhidas de forma aleatória por Bob para a leitura dos Fótons	|A>	|A>	|B>	|B>	|A>	|B>	|B>	|B>
Comparação entre Bases escolhidas por Alice e Bob	**C**	**E**	**C**	**E**	**C**	**C**	**C**	**E**
Valores definidos por Alice	**1**	**X**	**0**	**X**	**0**	**0**	**0**	**X**
Valores definidos por Bob	**1**	**X**	**1**	**X**	**0**	**0**	**1**	**X**

Tabela 12 – Comparação entre escolhas de Base e Chave gerada para Alice e Bob.

Nesse momento, Alice tem em sua posse a seguinte chave gerada: *"10000"*. Já Bob tem outra chave, que é: *"11001"*. Como se pode perceber, as

chaves geradas são diferentes. Nesse ponto na situação anterior, as chaves eram as mesmas, mas este fato ainda não é de conhecimento de Alice e Bob, o emissor e o receptor.

As medições de Bob foram afetadas pela intrusão de Eva. Essas intrusões podem provocar alterações, e a sua detecção é comprovada probabilisticamente. Para simplificar a simulação, mas sem perda de generalidade se Alice e Bob utilizam a mesma base, a probabilidade de Eva usar a mesma base vale 50% (se Alice e Bob utilizaram a Base A, por exemplo, a probabilidade de Eva também ter utilizado essa base é 50%). Agora, se Eva utiliza para monitorar os fótons a outra base, a probabilidade de Bob medir corretamente o valor do bit transmitido é de apenas 50% e não 100%, como deveria ser se não houvesse um espião ou se Eva tivesse optado pela base correta.

O próximo passo então é definir uma parte da chave gerada, para ser a chave que será compartilhada e através da comparação da chave compartilha identificar Eva, o intruso, no canal. Então Bob, como na situação anterior, define sua chave compartilhada como sendo as três primeiras posições da chave gerada, *"110"* e envia a Alice.

Alice, então, recebe e compara a chave compartilhada de Bob com a sua. Como a seguir:

Chave Compartilhada de Alice	1	0	0
Chave Compartilhada de Bob	1	1	0
Comparação entre Chaves compartilhadas	**C**	**E**	**C**

Tabela 13 - Comparação entre as chaves compartilhada de Alice e Bob.

Alice consegue então detectar uma diferença na chave gerada, então é identificada a presença de Eva, pois a presença de Eva resultou numa alteração de 33% da chave compartilhada. Alice então constata que a chave está comprometida. Esta então é descartada e uma nova tentativa deve ser

feita. Sempre que houver dúvidas sobre a confiabilidade da chave, esta deverá ser descartada e uma nova gerada.

Essa porcentagem de alteração não possui um valor fixo, se na simulação for considerado o meio ideal e sem perda de fótons, toda e qualquer alteração é atribuída a um intruso. Num meio real, existe uma certa interferência desse meio, impurezas do canal e possível perda de fótons, então define-se um percentual aceitável de erro.

3.3. Simulador

Definida as tecnologias e as suas funcionalidades, o simulador será exibido nas práticas, com as suas telas.

A seguir é exibida a tela inicial do simulador, onde se optará por uma das funcionalidades, “Transmissão Quântica” ou “Qkd – Protocolo BB84”:

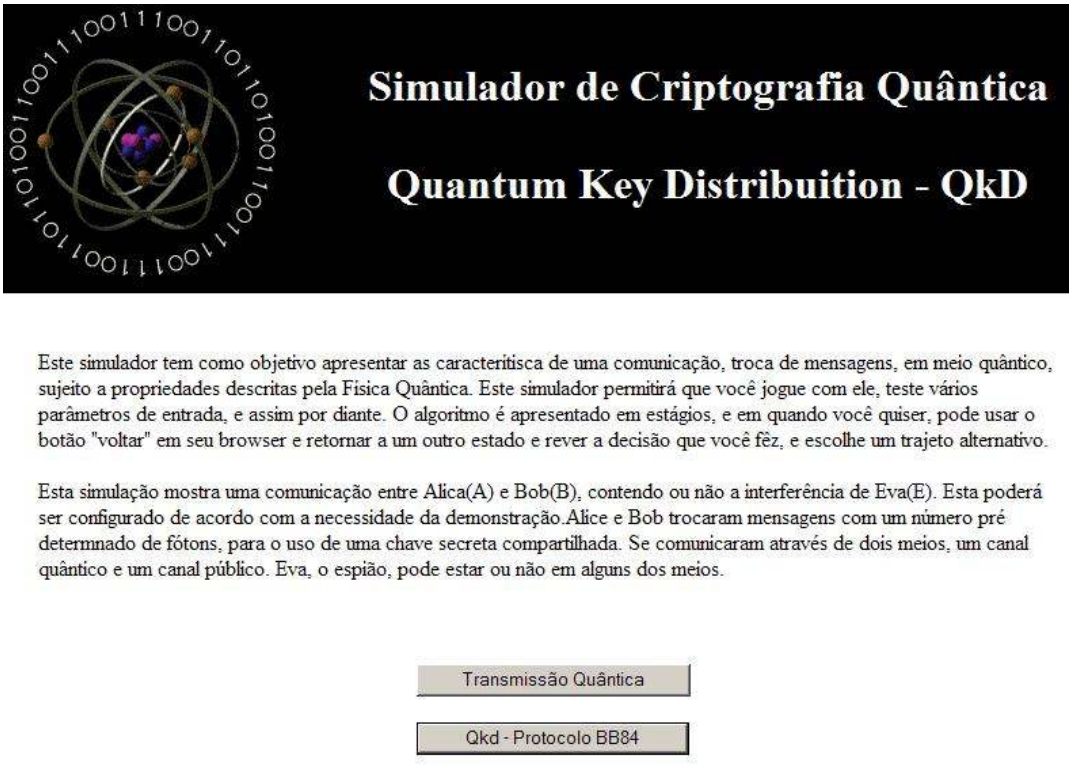

Figura 5 – Tela inicial do Simulador, escolha da funcionalidade

No primeiro momento será escolhida a funcionalidade “Transmissão Quântica”. A tela abaixo será exibida então, que é a tela de configuração, possibilitando algumas escolhas para a simulação

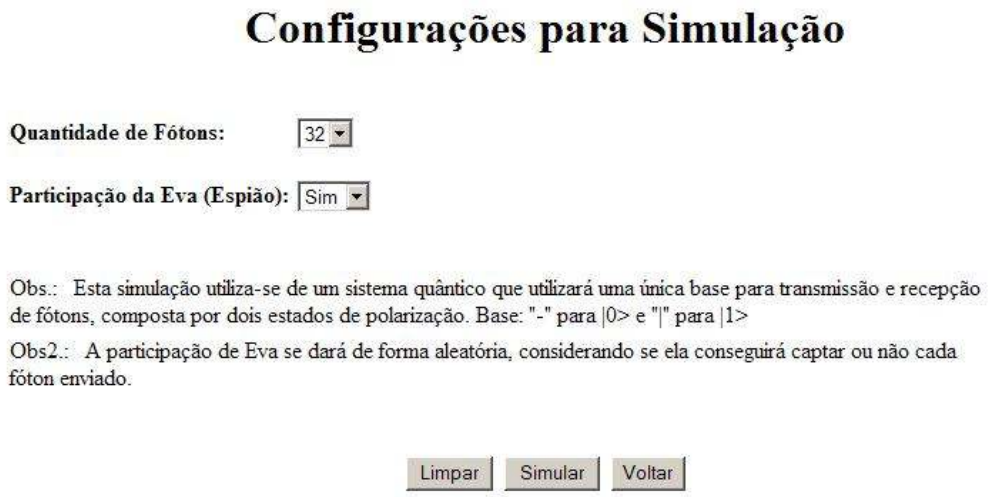

Figura 6 – Tela de configuração da Funcionalidade “Transmissão Quântica”

Configurada a simulação, então será dado prosseguimento, clicando no botão “Simular”.

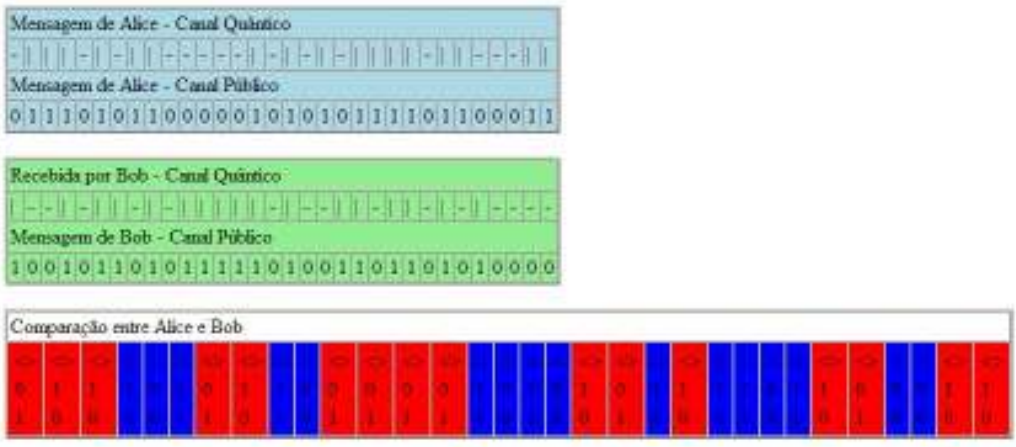

Figura 7 – Tela de simulação da Funcionalidade “Transmissão Quântica”

Na simulação acima, tem se três tabelas. Na primeira tabela, temos a mensagem que Alice deseja transmitir e a seqüência de base que foi usada para polarizar os fótons em acordo com a base definida. Já na segunda tabela, Tem-se a seqüência de base que Bob utilizou para a leitura dos fótons, e os resultados encontrados. A última tabela é a comparação entre o que foi enviado e o que foi lido, e a diferença é atribuída a interferência do espião. Não houve geração de chave, pois a intenção foi apenas exemplificar a comunicação no meio quântico.

Continuando a simulação, agora será executada a outra funcionalidade, a geração de chave criptográfica quântica. A tela a seguir, é a tela de configuração:

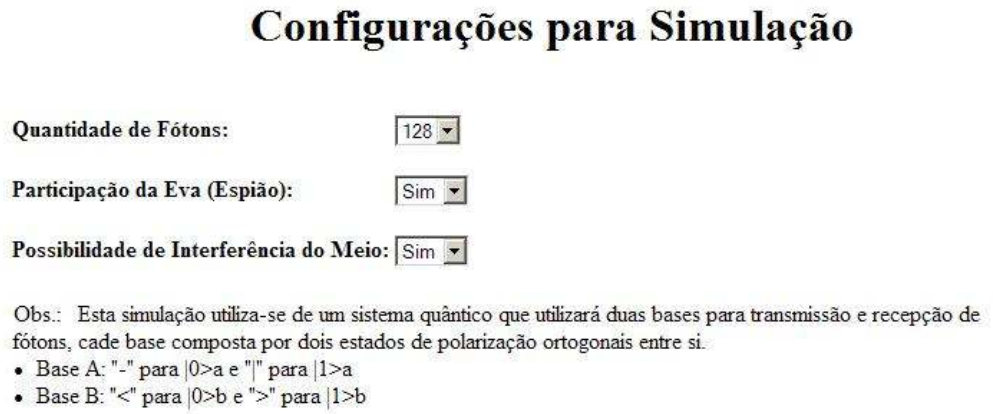

Figura 8 – Tela de configuração da Funcionalidade "Qkd – Protocolo BB84"

Nesta tela apresenta a possibilidade de interferência do meio ambiente na transmissão, essa interferência foi tratada como uma perda de fótons, e pode ser causada por campos eletro-magnético ou impurezas do meio de transmissão.

A tela seguintes apresentará então a simulação da geração da chave criptográfica quânica:

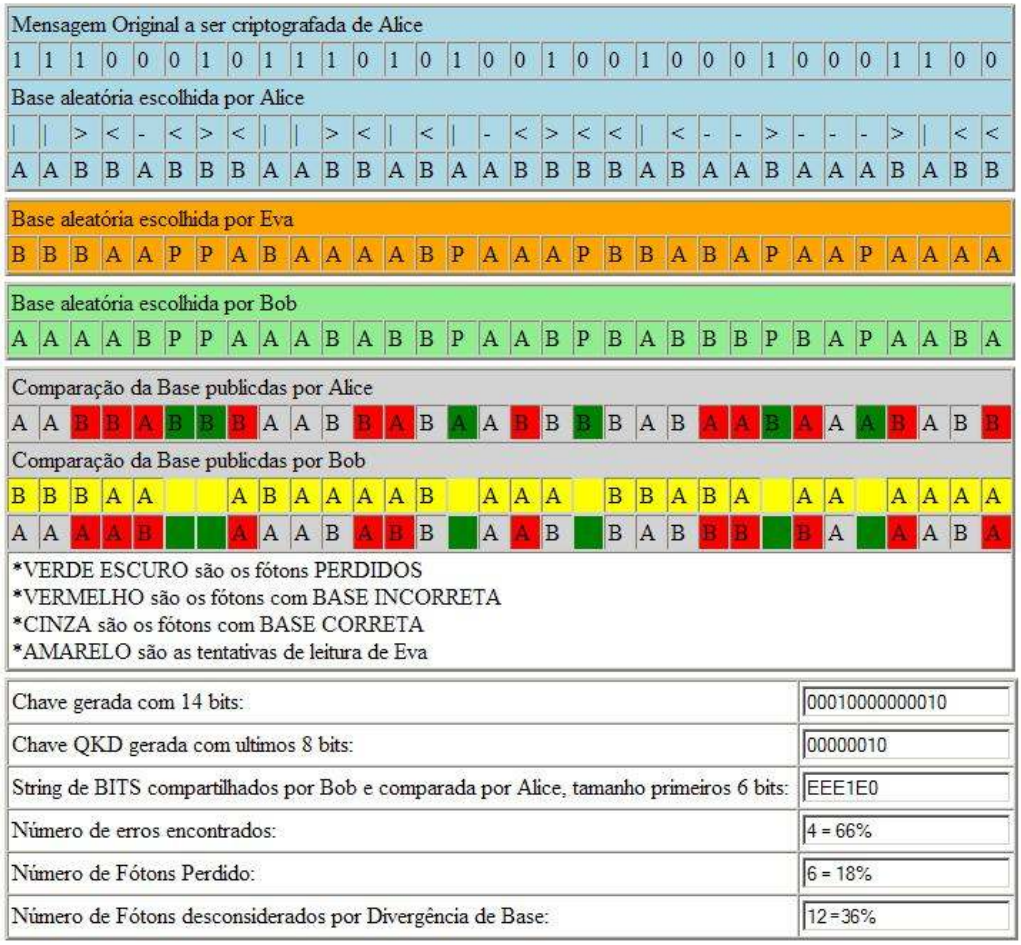

Figura 9 – Tela de simulação da Funcionalidade "Qkd – Protocolo BB84"

No exemplo acima, são mostrada quatro tabelas. A primeira é a geração de uma seqüência de bits de forma aleatória e sua conversão para os qubtis utilizando as bases definidas, ou seja polarização dos fótons que serão transmitidos. A escolha da base é feita de forma aleatória também.

Uma vez feita a transmissão, as duas tabelas seguintes representam, respectivamente, as tentativas de leituras de Eva e Bob, onde "A" e "B" são as bases escolhidas e "P" os fótons perdidos.

A tabela seguinte, e última, representa a comparação realizada por Alice e Bob, num primeiro momento se compara as escolhas de base, descartando-se as erradas. Num segundo momento, com a seqüência de bits resultantes

das escolhas corretas, compartilha-se uma parte para identificar a presença do espião, Eva, na transmissão.

No fim da tela, as informações extraídas da transmissão, onde se verifica a chave gerada, que da origem a chave compartilhada e a chave criptográfica. A comparação é mostrada para se identificar onde a interferência de Eva foi detectada. E um percentual de perdas de informação admitidas no protocolo. No exemplo, a chave criptográfica deveria ser descartada, pois a chave compartilhada mostrou que existe um espião, e que não se pode garantir a privacidade da chave. Não se sabe quanto da chave foi descoberta, porém não se pode garantir a segurança da mensagem com uma chave parcialmente segura.

4. Conclusão

As mudanças no cotidiano das pessoas no mundo inteiro são constantes incentivos para o desenvolvimento de novas técnicas. Antes, para uma transação financeira acontecer, ela teria que passar por todo um procedimento, uma papelada, para se tornar possível. Hoje, basta um clique no mouse. Dessa forma, a segurança dos meios virtuais passou de ser apenas um acessório para algo essencial. Tornou-se elemento indispensável para qualquer ação.

Diante disto, se pesquisar temas para realização deste trabalho, resolvem-se tentar dar um passo conjunto e a frente nas correntes que estão estudando este assunto - o da segurança na rede (internet). Ou seja, decidiu-se que esta monografia teria o objetivo de criar um simulador que permitisse a troca de mensagens entre usuários de maneira rápida, eficaz e segura. E, para isto, utilizaria-se grandes pilares científicos atuais – a física quântica e a criptografia quântica.

O protótipo desenvolvido apresenta um estudo sobre a tecnologia de Criptografia Quântica - conceitos, aplicação e o desenvolvimento de uma ferramenta, que pode ser utilizada como instrumento didático de demonstração das características da QKD num computador convencional. Um tema que desperta muito interesse por trabalhar com a Física Quântica, e tudo que esta pode trazer de benefícios a várias áreas da ciência, como na computação, na biotecnologia, na nanotecnologia e na própria criptografia.

Uma das conclusões deste estudo é que, apesar de não se tratar de um tema novo, embora pouco estudado, é extremamente atual. É um assunto que se encontra sendo muito debatido no meio acadêmico, visto as discussões e pesquisas que ainda existem. Isto acontece, principalmente, pelo fato que muitas das tecnologias atuais, além da Criptografia Quântica, são baseadas nos conceitos de Física Quântica, que foi inaugurada no começo do século passado, em 1900, pelo físico alemão Max Planck, que foi o responsável por abrir caminho para a teoria Quântica.

No meio de tantas transações em meios públicos e com a tendência de que isso só venha a aumentar, a criptografia tornou-se um tema bastante

discutido atualmente, uma vez que é necessário melhorar a segurança na comunicação.

Grandes investimentos têm sido feito na área e várias técnicas têm sido usadas e outras tantas desenvolvidas, mas todas elas apresentam limitações pelo fato da maioria possuir obstáculos na hora da transmissão - no momento de compartilhar-se a chave criptográfica. Outras técnicas, por terem a geração da chave baseadas em métodos matemáticos, podem ser desfeitas, quebradas, usando-se um grande poder computacional.

Diante destes fatos, a Criptografia Quântica oferece um novo nível de segurança, que nenhuma das técnicas atuais oferece: um meio seguro de transmissão, uma vez que, se houver algum espião, será possível detectá-lo. Entretanto, como ainda é um meio caro e por poder somente detectar intruso apenas depois de se transmitir algo, ele é utilizado apenas para transmitir a chave criptográfica, depois de assegurado que o canal quântico esteja seguro. Dessa forma, ao idealizar o simulador deste trabalho, resolveu-se demonstrar uma maneira rápida, simples e barata de simular a implementação de um protocolo de distribuição de chaves quânticas, de uma forma didática.

A técnica de geração de chaves, por meio de números aleatórios, ou seja, chaves aleatórias, vem a incrementar ainda mais a segurança da Criptografia Quântica, pois se baseia nas leis da natureza e, atualmente, não há nenhuma experiência que possa contestar a validade da Mecânica Quântica e, por isso, invalidar a Criptografia Quântica.

Outra conclusão, que representa um objetivo inicial deste trabalho atingido, é que a distribuição de chave Quântica foi simulada com sucesso – que demonstrou como seria gerada uma chave *QKD* - e, com isso, provou-se que o canal de comunicação é seguro.

Conclui-se também que, para saber se existe algum intruso no canal de comunicação, faz-se necessário uma transmissão, seja de chave ou uma seqüência qualquer, e, por meio dessa transmissão, pode-se medir se houve interferência ou não comparando os resultados da mesma. Porém, a comunicação Quântica ainda possui limitações, como taxa de transferência baixa e curtas distâncias alcançadas e, por isso, torna-se inapropriada para a transmissão pura de dados.

Este fato não a torna inválida para a transmissão de chaves, pois, uma vez a chave transmitida, o receptor tem condições de checar se ela foi lida ou alterada e, caso isso tenha ocorrido, ele invalida a chave e pede uma retransmissão de uma nova chave.

A utilização do protocolo BB84 nesse processo de criação e transmissão da chave agrega, além da segurança citada acima, o processo de geração do seqüencial de bits transmitido para a geração de chave. Este é feito de forma aleatória, pois a chave dependerá dos filtros escolhidos, de forma aleatória por Alice e Bob, aumentando ainda mais a segurança.

A chave resultante dessa operação não é definida por ninguém, é conseqüência das escolhas indiretas de Alice e Bob, como mostrado na simulação. Mesmo que Alice mande sempre a mesma seqüência de bits, as escolhas aleatórias de Bob gerariam uma chave diferente.

Em geral, o tamanho da chave depende do tamanho da seqüência de bits transmitida, do número de bases adotadas entre Alice e Bob, pois probabilidade de Bob acertar o filtro incidirá diretamente nos bits aproveitados ou descartados para a geração da nova chave, e no tamanho da chave compartilhada.

Um canal de comunicação seguro entre dois pólos foi criado na implementação das técnicas aqui estudadas e analisadas de maneira mais simples que muitos estudos demonstram. Além disso, esta monografia representa um primeiro passo para base de trabalhos futuros, pois fornece dados e material para que o desenvolvimento desta técnica continue e, com isso, novos protocolos sejam criados por meio do simulador com o objetivo de comparar suas vantagens e desvantagens, gerar um novo protocolo de Criptografia Quântica, ou criar uma aplicação, que por meio da geração automatizada de uma chave Quântica, cifre, transmita e decifre a mensagem.

Referências Bibliográficas

[1]. ALVES, Flávio Luis. “Computação Quântica: Fundamentos Físicos e Perspectivas”, Univ. Federal de Lavras. Lavras, MG, 2003;

[2]. BENNETT, C.H. BRASSARD, G. in Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing. Bangalore, India, 1984, 175-179.

[3]. BENNETT, C.H., Phys. Rev. Lett. 68, 3121, 1992.

[4]. BUCHMANN, Johannes A. “Introdução a Criptografia”. Ed. Berkeley, São Paulo 2002.

[5]. CROCA, J.R.. “A razão na Física Quântica”, Portugal, 2004;

[6]. DAVIDOVICH, Luiz. “O Mundo dos Quantas: de Planck e Eisntein ao Computador Quântico”. Univ. Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2005;

[7]. GILMORE, Robert. “Alice no País do Quantum”. Ed. Jorge Zahar, Rio de Janeiro, 1998.

[8]. OLIVIERA, Anderson Gomes de. “Criptografia usando protocolos quânticos”, Univ. Federal de Lavras. Lavras, MG, 2004.

[9]. PESSOA JÚNIOR, Osvaldo. “Conceitos de Física Quântica”, Ed. Livraria da Física, São Paulo, SP, 2003.

[10]. RIGOLIN, Gustavo. RIEZNIK, Andrés Aníbal. “Introdução à Criptografia Quântica”, UNICAMP. Campinas, SP, 2005.

[11]. SINGH, Simon. “O livro dos Códigos”, Rio de Janeiro, Ed. Record, 2001.

[12]. STIX, Gary. “Os Segredos mais bem guardados”, pág.38 a 45, Scientific American Brasil, Fev/2005.

[13]. SULZBACH, Jaime André. “Análise de viabilidade de Criptografia Quântica”, Univ. Do Vale do Rio dos Sinos, São Leopoldo, RS, 2003;

[14]. TAKAGI, Nilton Hideki. “Fundamentos Matemáticos da Criptografia Quântica”, Univ. Federal de Mato Grosso, Cuiabá, MT, 2003;

[15]. XAVIER, Guilherme Barreto. “Esquemas de modulação para distribuição Quântica de chaves com codificação de freqüência”. UFRJ, Rio de Janeiro, 2005;

[16]. http://agenciact.mct.gov.br/, disponível em 09/2006;

[17]. http://www.inovacaotecnologica.com.br/noticias/meta.php?meta=Criptografia, acessado em 09/2006;

[18]. http://www.wikipedia.org/wiki/, acessado em 09/2006;

[19]. http://www.research.ibm.com/people/b/bennetc/chbbib.htm, acessado em 11/2006;

a) Anexos

Anexo A - Algoritmo PRIMES

Input: integer $n > 1$

1. if (n is of the form a^b , $b > 1$) output COMPOSITE;
2. $r = 2$;
3. while ($r < n$) {
4. if (gcd(n,r) $\neq 1$) output COMPOSITE;
5. if (r is prime)
6. let q be the largest prime factor of r - 1;
7. if f ($q \geq 4\sqrt{r}\log n$) and ($n^{\frac{r-1}{q}} \neq 1 (\mathrm{mod}\, r)$)
8. break;
9. r r + 1;
10. }
11. for a=1 to $2\sqrt{r}\log n$
12. if ($(x - a)^n (x^n - a) \pmod{x^r - 1, n}$) output COMPOSITE;
13. output PRIME;

Obs.: A função gcd (Greatest Common Divisor) retorna o máximo divisor comum de dois inteiros.

Anexo B – Código fonte do Simulador

```
<!--
Página inicial do sistema
Oferece as opções Trasnimissão Quantica / BB84
-->
<!-- index.jsp -->
<%@page import="java.util.*"%>
<html>
  <head>
    <title>Simulador de Criptografia Quântica</title>
    <script>
          function bb84(){
                form.action = "<%=
request.getContextPath()%>/jsp/indexQkd.jsp";
                form.submit();
          }
    </script>
  </head>
  <%@ include file="header.jsp" %>
    <table align="left">
    <form name="form" action="<%= request.getContextPath()%>/jsp/conf.jsp">
      <tr>
        <td width="20"> </td>
        <td width="760">
          <br><br><br>
          <br><br><br>
        </td>
        <td width="20"> </td>
      </tr>
       <tr>
        <td width="20"> </td>
        <td width="760" align="center">
```

```
                <br>
                <br>
                <input type="submit" value="Transmissão Quântica">
                <br><br>
                <input type="button" value=" Qkd - Protocolo BB84 "
onclick="bb84();">
                <br><br>
              </td>
              <td width="20"> </td>
             </tr>
            </form>
            </table>
            <%//@ include file="footer.jsp" %>
          </body>
        </html>



        <!--
        Configurações para<br>Simulação da Transmissão Quântica
        -->
        <!-- conf.jsp -->
        <%@page import="java.util.*"%>
        <% int rd = (new Random()).nextInt(50); %>
        <html>
          <head>
            <title>Simulador de Criptografia Quântica</title>
          </head>
            <script>
                function irQkd(){
                        document.form.action                =                "<%=
request.getContextPath()%>/confQKD";
                        document.form.submit();
```

```
    }
</script>
    <%@ include file="header.jsp" %>
<form name="form" action="<%= request.getContextPath()%>/conf">
<input type="hidden" name="fase" value="0">
<table align="left">
 <tr>
  <td width="20"> </td>
  <td width="760" align="center">
    <br>
    <h1>Configurações para<br>Simulação da Transmissão Quântica</h1>
    <br>
  </td>
  <td width="20"> </td>
 </tr>
 <tr>
  <td width="20"> </td>
  <td width="760">
   <table border="0" cellpadding="2" cellspacing="2">
    <tr>
     <td><h4>Quantidade de Fótons:</h4></td>
     <td align="left" valign="top">
      <select name="tamanho">
       <option value="8">08</option>
       <option value="16">16</option>
       <option value="32" selected>32</option>
       <option value="64">64</option>
      </select>
     </td>
    </tr>
    <tr>
     <td><h4>Participação da Eva (Espião):</h4></td>
     <td align="left" valign="top">
```

```
          <select name="participacao">
            <option value="true">Sim</option>
            <option value="false">Não</option>
          </select>
        </td>
      </tr>
    </table>
    <br>
    <table width="700" cellpadding="2" cellspacing="2">
      <tr>
        <td align="center">
          <hr>
          <h3>Base Definida: "-" para |0> e "|" para |1></h3>
          <hr>
        </td>
      </tr>
    </table>
  </td>
  <td width="20"> </td>
 </tr>
  <tr>
  <td width="20"> </td>
  <td width="760" align="center">
      <br>
      <br>
      <input type="reset" value="Limpar"> 
      <input type="submit" value="Simular"> 
      <input type="button" value="Voltar" onclick="history.back();">
  </td>
  <td width="20"> </td>
 </tr>
</table>
</form>
```

```
  </body>
</html>



<!--
Exibi a transmissão quantica com espião
-->
<!-- simulacaoSpy.jsp -->
<!-- Configurações para<br>Simulação da Transmissão Quântica  -->
<%@page import="java.util.*,br.ceub.scq.bean.*"%>
<%ArrayList arrayMensagem = (ArrayList) session
        .getAttribute("arrayMensagem");
    ArrayList arrayBob = (ArrayList) session.getAttribute("arrayBob");
    String flag = request.getParameter("flag") == null ? "0" : request
        .getParameter("flag");
%>
<html>
<head>
<title>Simulador de Criptografia Quântica</title>
</head>
<%@ include file="header.jsp"%>
<script>
    function continuar(){
        aux = parseInt(document.form.flag.value);
        document.form.flag.value = aux+1;
        document.form.submit();
    }
    function resetar(){
        document.form.action = "<%=
request.getContextPath()%>/limpar";
        document.form.submit();
    }
```

```
  </script>
 <form name="form" action="<%= request.getContextPath()%>/conf"><input type="hidden" name="flag" value="<%= flag %>"> <input type="hidden"
  name="tamanho" value="<%= request.getParameter("tamanho") %>"> <input type="hidden" name="participacao"
  value="<%= request.getParameter("participacao") %>"> <br>
 <br>
 <table align="left">
  <tr>
   <td width="20"> </td>
   <td width="760" align="center">
   <table border="2" cellpadding="2" cellspacing="0" bgcolor="lightblue">
    <tr>
     <td colspan="<%= arrayMensagem.size() %>"><b>Mensagem de Alice - Canal Quântico</b></td>
    </tr>
    <tr>
     <%for (Iterator it = arrayMensagem.iterator(); it.hasNext();) {
          MensagemQuantica bean = (MensagemQuantica) it.next();%>
     <td><b><%=bean.getSimboloMensagem()%></b></td>
     <%} %>
    </tr>
    <tr>
     <td colspan="<%= arrayMensagem.size() %>"><b>Mensagem de Alice - Canal Público</b></td>
    </tr>
    <tr>
     <%for (Iterator it = arrayMensagem.iterator(); it.hasNext();) {
          MensagemQuantica bean = (MensagemQuantica) it.next();%>
     <td><b><%=bean.getBitMensagem()%></b></td>
     <%} %>
    </tr>
   </table>
```

```
<%if (!flag.equals("0")) {

        %> <br>
<table border="2" cellpadding="2" cellspacing="0" bgcolor="lightgreen">
 <tr>
  <td colspan="<%= arrayBob.size() %>"><b>Recebida por Bob - Canal Quântico   </b></td>
 </tr>
 <tr>
  <%for (Iterator it = arrayBob.iterator(); it.hasNext();) {
        MensagemQuantica bean = (MensagemQuantica) it.next();%>
  <td><b><%=bean.getSimboloMensagem()%></b></td>
  <%} %>
 </tr>
 <tr>
  <td colspan="<%= arrayBob.size() %>"><b>Mensagem de Bob - Canal Público</b></td>
 </tr>
 <tr>
  <%for (Iterator it = arrayBob.iterator(); it.hasNext();) {
        MensagemQuantica bean = (MensagemQuantica) it.next();%>
  <td><b><%=bean.getBitMensagem()%></b></td>
  <%} %>
 </tr>
</table>
<br>
<table border="2" cellpadding="2" cellspacing="0" bgcolor="#FFFFFF">
 <tr>
  <td colspan="<%= arrayMensagem.size() %>"><b>Comparação entre Alice e Bob</b></td>
 </tr>
 <tr>
  <%for (int i = 0; i < arrayMensagem.size(); i++) {
```

```
                    MensagemQuantica beanAlice = (MensagemQuantica)
arrayMensagem
                        .get(i);
                    MensagemQuantica beanBob = (MensagemQuantica) arrayBob
                        .get(i); %>
                <td                                                    bgcolor="<%=
beanAlice.isIgualMensagem(beanBob.getBitMensagem())?"blue":"red"%>">
                <table>
                  <tr>
                    <td><b><font
color="white"><%=beanAlice.isIgualMensagem(beanBob.getBitMensagem()) ? "=" :
"x"%></font></b></td>
                  </tr>
                  <tr>
                    <td><b><font
color="white"><%=beanAlice.getBitMensagem()%></font></b></td>
                  </tr>
                  <tr>
                    <td><b><font
color="white"><%=beanBob.getBitMensagem()%></font><b></td>
                  </tr>
                </table>
                </td>
                <%} %>
              </tr>
            </table>
            <%} %></td>
            <td width="20"> </td>
          </tr>
          <tr>
            <td width="20"> </td>
            <td width="760" align="center"><br>
            <br>
```

```
        <input type="button" value="Limpar"
onclick="resetar();">  <input type="button" value="Continuar"
onclick="continuar();">  <input
          type="button" value="Voltar" onclick="history.back();"></td>
        <td width="20"> </td>
      </tr>
    </table>
    </form>
    </body>
    </html>



    <!--
    Página de Erro Inesperado.
    -->
    <!-- erro.jsp -->
    <%@page import="java.util.*"%>
    <%// int rd = (new Random()).nextInt(50); %>
    <html>
      <head>
        <title>Simulador de Criptografia Quântica</title>
      </head>
      <body leftmargin="0" topmargin="0">
        <form action="conf.jsp">
        <table align="left">
          <tr>
            <td width="20"> </td>
            <td width="760" align="center">
              <br>
              <font color=red>Erro</font>
              <br>
            </td>
```

```
        <td width="20"> </td>
       </tr>
      </table>
      </form>
     </body>
    </html>



    <!--
    Página de Cabeçalho, ela é exibida junto com todas as outras páginas.
    -->

    <!-- INICIO header.jsp -->
    <body leftmargin="0" topmargin="0">
     <table bgcolor="black" width="800" border="4" cellpadding="0"
cellspacing="0" bordercolor="#000000">
      <tr>
       <td align="center" valign="middle">
        <img src="<%= request.getContextPath()%>/img/molecula binaria.bmp">
       </td>
       <td width="20"> </td>
       <td align="center" valign="middle">
        <h1>
        <font color="white">Simulador do Protocolo Quântico
BB84<br><br>Quantum Key Distribuition - QKD</font>
        </h1>
       </td>
      </tr>
     </table>
    <!-- FIM header.jsp -->
```

```
<!--
Página de Configuração da Simulação do BB84
-->

<!-- indexQkd.jsp -->
<%@page import="java.util.*"%>
<html>
 <head>
  <title>Simulador de Criptografia Quântica</title>
 </head>
        <%@ include file="header.jsp" %>
  <form action="<%= request.getContextPath()%>/confQKD">
  <input type="hidden" name="fase" value="0">
  <table align="left">
   <tr>
    <td colspan="3"> </td>
   </tr>
   <tr>
    <td colspan="3">
     <table align="left">
      <tr>
       <td width="20"> </td>
       <td width="760" align="center">
          <br>
          <h1>Configurações para Simulação de Geração de<br>Chaves Criptográficas utilizando o<br>Protocolo Quântico BB84</h1>
          <br><br>
       </td>
       <td width="20"> </td>
      </tr>
      <tr>
       <td width="20"> </td>
```

```
<td width="760">
  <table>
    <tr>
      <td><h4>Quantidade de Fótons:</h4></td>
      <td valign="top">
        <select name="tamanho">
          <option value="32">32</option>
          <option value="64">64</option>
          <option value="128" selected>128</option>
          <option value="256">256</option>
          <option value="384">384</option>
          <option value="512">512</option>
        </select>
      </td>
    </tr>
    <tr>
      <td><h4>Participação da Eva (Espião):</h4></td>
      <td valign="top">
        <select name="participacao">
          <option value="0">Não</option>
          <option value="1" selected>Sim</option>
        </select>
      </td>
    </tr>
    <tr>
      <td><h4>Possibilidade de Interferência do Meio:</h4></td>
      <td valign="top">
        <select name="hasPerda">
          <option value="0">Não</option>
          <option value="1" selected>Sim</option>
        </select>
      </td>
    </tr>
```

```
                    </table>
                    <table width="700" cellpadding="2" cellspacing="2">
                      <tr>
                        <td>
                          <hr>
                          <h3>Bases Definidas:
                          <br>
                            <li>Base A: "-"  para
|0>a  e  "|"  para |1>a
                            <li>Base B: "<" para
|0>b  e  ">" para |1>b</h3>
                          <hr>
                        </td>
                      </tr>
                    </table>
                   <td width="20"> </td>
                  </tr>
                 </table>
               </td>
              </tr>
               <tr>
                <td width="20"> </td>
                <td width="760" align="center">
                  <br>
                  <br>
                  <input type="submit" value="Continuar">
                </td>
                <td width="20"> </td>
              </tr>
            </table>
            </form>
          </body>
        </html>
```

```
<!--
Página da Transmissão Quântica sem Espião
-->
<!-- simulacaoNoSpy.jsp -->
<%@page import="java.util.*, br.ceub.scq.bean.*"%>
<%
ArrayList arrayMensagem = (ArrayList)session.getAttribute("arrayMensagem");
ArrayList arrayBob      = (ArrayList)session.getAttribute("arrayBob");
String flag = request.getParameter("flag")==null?"0":request.getParameter("flag");
%>
<html>
  <head>
    <title>Simulador de Criptografia Quântica</title>
  </head>
 <%@ include file="header.jsp" %>
 <script>
      function continuar(){
            aux = parseInt(document.form.flag.value);
            document.form.flag.value = aux+1;
            document.form.submit();
      }
      function resetar(){
            document.form.action = "<%= request.getContextPath()%>/limpar";
            document.form.submit();
      }
 </script>
  <form name="form" action="<%= request.getContextPath()%>/conf">
  <input type="hidden" name="flag" value="<%= flag %>">
```

```
        <input type="hidden" name="tamanho" value="<%= request.getParameter("tamanho") %>">
        <input type="hidden" name="participacao" value="<%= request.getParameter("participacao") %>">
        <br><br>
        <table align="left">
          <tr>
            <td width="20"> </td>
            <td width="760" align="center">
              <table border="2" cellpadding="2" cellspacing="0" bgcolor="lightblue">
                <tr>
                  <td colspan="<%= arrayMensagem.size() %>"><b>Mensagem de Alice - Canal Quântico</b></td>
                </tr>
                <tr>
                  <%for(Iterator it = arrayMensagem.iterator(); it.hasNext();){
                        MensagemQuantica bean = (MensagemQuantica)it.next();%>
                  <td><b><%= bean.getSimboloMensagem()%></b></td>
                  <%} %>
                </tr>
                <tr>
                  <td colspan="<%= arrayMensagem.size() %>"><b>Mensagem de Alice - Canal Público</b></td>
                </tr>
                <tr>
                  <%for(Iterator it = arrayMensagem.iterator(); it.hasNext();){
                        MensagemQuantica bean = (MensagemQuantica)it.next();%>
                  <td><b><%= bean.getBitMensagem()%></b></td>
                  <%} %>
                </tr>
                </table>
```

```
<% if(!flag.equals("0")){ %>
<br>
<table border="2" cellpadding="2" cellspacing="0"
bgcolor="lightgreen">
<tr>
 <td colspan="<%= arrayBob.size() %>"><b>Recebida por Bob - Canal
Quântico   </b></td>
</tr>
<tr>
 <%for(Iterator it = arrayBob.iterator(); it.hasNext();){
                MensagemQuantica bean =
(MensagemQuantica)it.next();%>
 <td><b><%= bean.getSimboloMensagem()%></b></td>
 <%} %>
</tr>
<tr>
 <td colspan="<%= arrayBob.size() %>"><b>Mensagem de Bob - Canal
Público</b></td>
</tr>
<tr>
 <%for(Iterator it = arrayBob.iterator(); it.hasNext();){
                MensagemQuantica bean =
(MensagemQuantica)it.next();%>
 <td><b><%= bean.getBitMensagem()%></b></td>
 <%} %>
</tr>
</table>
<br>
<table border="2" cellpadding="2" cellspacing="0"
bgcolor="#FFFFFF">
 <tr>
  <td colspan="<%= arrayMensagem.size() %>"><b>Comparação entre
Alice e Bob</b></td>
```

```
                </tr>
                <tr>
                  <%for(int i = 0; i < arrayMensagem.size(); i++){
                                MensagemQuantica beanAlica =
(MensagemQuantica)arrayMensagem.get(i);
                                MensagemQuantica beanBob =
(MensagemQuantica)arrayBob.get(i); %>
                  <td bgcolor="<%=
beanAlica.isIgualMensagem(beanBob.getBitMensagem())?"blue":"red"%>">
                   <b><font color="white"><%=
beanAlica.isIgualMensagem(beanBob.getBitMensagem())?"=":"x"%></font></b>
                  </td>
                  <%} %>
                  </tr>
                 </table>
                 <%} %>
               </td>
               <td width="20"> </td>
              </tr>
              <tr>
               <td width="20"> </td>
               <td width="760" align="center">
                 <br>
                 <br>
                 <input type="button" value="Limpar"
onclick="resetar();">  <input type="button" value="Continuar"
onclick="continuar();">  <input type="button" value="Voltar"
onclick="history.back();">
               </td>
               <td width="20"> </td>
              </tr>
             </table>
             </form>
```

```
  </body>
</html>



<!--
Página da Transmissão Quântica com Espião
-->
<!-- simulacaoSpy.jsp -->
<%@page import="java.util.*,br.ceub.scq.bean.*"%>
<%ArrayList arrayMensagem = (ArrayList) session
                .getAttribute("arrayMensagem");
        ArrayList arrayBob = (ArrayList) session.getAttribute("arrayBob");
        String flag = request.getParameter("flag") == null ? "0" : request
                .getParameter("flag");
%>
<html>
<head>
<title>Simulador de Criptografia Quântica</title>
</head>
<%@ include file="header.jsp"%>
<script>
        function continuar(){
                aux = parseInt(document.form.flag.value);
                document.form.flag.value = aux+1;
                document.form.submit();
        }
        function resetar(){
                document.form.action = "<%=
request.getContextPath()%>/limpar";
                document.form.submit();
        }
  </script>
```

```
<form name="form" action="<%= request.getContextPath()%>/conf"><input type="hidden" name="flag" value="<%= flag %>"> <input type="hidden"
  name="tamanho" value="<%= request.getParameter("tamanho") %>"> <input type="hidden" name="participacao"
  value="<%= request.getParameter("participacao") %>"> <br>
<br>
<table align="left">
  <tr>
    <td width="20"> </td>
    <td width="760" align="center">
    <table border="2" cellpadding="2" cellspacing="0" bgcolor="lightblue">
      <tr>
        <td colspan="<%= arrayMensagem.size() %>"><b>Mensagem de Alice - Canal Quântico</b></td>
      </tr>
      <tr>
        <%for (Iterator it = arrayMensagem.iterator(); it.hasNext();) {
            MensagemQuantica bean = (MensagemQuantica) it.next();%>
        <td><b><%=bean.getSimboloMensagem()%></b></td>
        <%} %>
      </tr>
      <tr>
        <td colspan="<%= arrayMensagem.size() %>"><b>Mensagem de Alice - Canal Público</b></td>
      </tr>
      <tr>
        <%for (Iterator it = arrayMensagem.iterator(); it.hasNext();) {
            MensagemQuantica bean = (MensagemQuantica) it.next();%>
        <td><b><%=bean.getBitMensagem()%></b></td>
        <%} %>
      </tr>
    </table>
    <%if (!flag.equals("0")) {
```

```
            %> <br>
        <table border="2" cellpadding="2" cellspacing="0" bgcolor="lightgreen">
         <tr>
          <td colspan="<%= arrayBob.size() %>"><b>Recebida por Bob - Canal Quântico   </b></td>
         </tr>
         <tr>
          <%for (Iterator it = arrayBob.iterator(); it.hasNext();) {
                MensagemQuantica bean = (MensagemQuantica) it.next();%>
          <td><b><%=bean.getSimboloMensagem()%></b></td>
          <%} %>
         </tr>
         <tr>
          <td colspan="<%= arrayBob.size() %>"><b>Mensagem de Bob - Canal Público</b></td>
         </tr>
         <tr>
          <%for (Iterator it = arrayBob.iterator(); it.hasNext();) {
                MensagemQuantica bean = (MensagemQuantica) it.next();%>
          <td><b><%=bean.getBitMensagem()%></b></td>
          <%} %>
         </tr>
        </table>
        <br>
        <table border="2" cellpadding="2" cellspacing="0" bgcolor="#FFFFFF">
         <tr>
          <td colspan="<%= arrayMensagem.size() %>"><b>Comparação entre Alice e Bob</b></td>
         </tr>
         <tr>
          <%for (int i = 0; i < arrayMensagem.size(); i++) {
```

```
                    MensagemQuantica beanAlice = (MensagemQuantica)
arrayMensagem
                        .get(i);
                    MensagemQuantica beanBob = (MensagemQuantica) arrayBob
                        .get(i); %>
                <td                                                    bgcolor="<%=
beanAlice.isIgualMensagem(beanBob.getBitMensagem())?"blue":"red"%>">
                <table>
                  <tr>
                    <td><b><font
color="white"><%=beanAlice.isIgualMensagem(beanBob.getBitMensagem()) ? "=" :
"x"%></font></b></td>
                  </tr>
                  <tr>
                    <td><b><font
color="white"><%=beanAlice.getBitMensagem()%></font></b></td>
                  </tr>
                  <tr>
                    <td><b><font
color="white"><%=beanBob.getBitMensagem()%></font><b></td>
                  </tr>
                </table>
                </td>
                <%} %>
              </tr>
            </table>
            <%} %></td>
            <td width="20"> </td>
          </tr>
          <tr>
            <td width="20"> </td>
            <td width="760" align="center"><br>
            <br>
```

```
        <input type="button" value="Limpar"
onclick="resetar();">  <input type="button" value="Continuar"
onclick="continuar();">  <input
          type="button" value="Voltar" onclick="history.back();"></td>
        <td width="20"> </td>
      </tr>
    </table>
    </form>
    </body>
    </html>



    <!--
    Arquivo de configuração do servidor de aplicação (Tomcat)
    -->

    <?xml version="1.0" encoding="ISO-8859-1"?>
    <!-- web.xml -->
    <!DOCTYPE web-app
        PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
       "http://java.sun.com/dtd/web-app_2_3.dtd">
    <web-app>
            <!-- servlets    -->
       <servlet>
          <servlet-name>conf</servlet-name>
          <servlet-class>br.ceub.scq.controle.Configuracao</servlet-class>
          <load-on-startup>3</load-on-startup>
       </servlet>
       <servlet>
          <servlet-name>confQKD</servlet-name>
          <servlet-class>br.ceub.scq.controle.ConfiguracaoQKD</servlet-class>
          <load-on-startup>4</load-on-startup>
```

```
    </servlet>
    <servlet>
       <servlet-name>limpar</servlet-name>
       <servlet-class>br.ceub.scq.controle.Resetar</servlet-class>
       <load-on-startup>5</load-on-startup>
    </servlet>
          <!-- fim de servlet    -->

          <!-- Mapeamento       -->
    <servlet-mapping>
       <servlet-name>confQKD</servlet-name>
       <url-pattern>/confQKD</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
       <servlet-name>conf</servlet-name>
       <url-pattern>/conf</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
       <servlet-name>limpar</servlet-name>
       <url-pattern>/limpar</url-pattern>
    </servlet-mapping>
          <!-- Fim Mapeamento   -->

    <session-config>
       <session-timeout>30</session-timeout>
    </session-config>
    <welcome-file-list>
       <welcome-file>/jsp/index.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

```
/*
Classe de Bean BaseQauntica
*/
package br.ceub.scq.bean;

public class BaseQuantica {

	private char tipoBase;
	private char bit1;
	private char bit0;



	public BaseQuantica(char tipoBase) {
		super();
		if(tipoBase == 'A'){
			this.bit1 = '|';
			this.bit0 = '-';
		} else {
			this.bit1 = '>';
			this.bit0 = '<';
		}
		this.tipoBase = tipoBase;
	}

	public BaseQuantica() {
		super();
	}

	public char getBit0() {
		return bit0;
	}
```

```
	public void setBit0(char bit0) {
		this.bit0 = bit0;
	}

	public char getBit1() {
		return bit1;
	}

	public void setBit1(char bit1) {
		this.bit1 = bit1;
	}

	public char getTipoBase() {
		return tipoBase;
	}

	public void setTipoBase(char tipoBase) {
		this.tipoBase = tipoBase;
	}

	public boolean isIgualBase(char tipoBase){
		if(getTipoBase() == tipoBase)
			return true;
		else
			return false;
	}
}
```

```
/**
Classe que faz a simulação quântica na transimssão
 */
package br.ceub.scq.controle;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import br.ceub.scq.bean.MensagemQuantica;

/**
 * @author cid
 *
 */
public class Configuracao extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private final String ERROR_PAGE = "/SimuladorCQ/jsp/erro.jsp";

	private String pagDestino = "";
	private HttpSession session;

	protected void service(HttpServletRequest req, HttpServletResponse res)
			throws ServletException, IOException {
```

```
                try {
                        int tamanho =
Integer.parseInt(req.getParameter("tamanho"));
                        boolean hasSpy =
req.getParameter("participacao").equals("false") ? false : true;

                        ArrayList arrayMensagem = new ArrayList();
                        Random rd = new Random();


        if(getSession(req).getAttribute("arrayMensagem")==null){
                                for (int i = 0; i < tamanho; i++) {
                                        MensagemQuantica bean = new
MensagemQuantica(String.valueOf(rd.nextInt(50)%2),String.valueOf(rd.nextInt(30)%2
));
                                        arrayMensagem.add(bean);
                                }
                        } else {
                                arrayMensagem =
(ArrayList)getSession(req).getAttribute("arrayMensagem");
                        }

                        ArrayList arrayBob = new ArrayList();
                        if (hasSpy){
                                for(int i = 0; i < arrayMensagem.size(); i++){
                                        boolean aux =
(rd.nextInt(10)%2)==0?true:false;
                                        MensagemQuantica bean =
(MensagemQuantica)arrayMensagem.get(i);
                                        MensagemQuantica bean2 = new
MensagemQuantica(bean.getBitBase(), bean.getBitMensagem());
                                        if(aux)
                                                bean2.inverteBitAll();
```

```
                    arrayBob.add(i,bean2);
                }
                pagDestino = "/jsp/simulacaoSpy.jsp";
            }else{
                arrayBob = (ArrayList)arrayMensagem.clone();
                pagDestino = "/jsp/simulacaoNoSpy.jsp";
            }

        getSession(req).setAttribute("arrayMensagem",arrayMensagem);
            getSession(req).setAttribute("arrayBob",arrayBob);

            RequestDispatcher red =
getServletContext().getRequestDispatcher(pagDestino);
            red.forward(req,res);

        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(e.getMessage());
            res.sendRedirect(ERROR_PAGE);
        }
    }

    private HttpSession getSession(HttpServletRequest req) {
        return session!=null?session:req.getSession();
    }
}
```

```
/*
Classe Bean que define uma mensagem quântica
*/
package br.ceub.scq.bean;

public class MensagemQuantica {

        private String bitMensagem;
        private String bitBase;
        private BaseQuantica base;
   private boolean isPerdido;

        public boolean isPerdido() {
      return isPerdido;
   }

   public void setPerdido(boolean isPerdido) {
      this.isPerdido = isPerdido;
   }

   public MensagemQuantica(String bit, char base, boolean isPerdido) {
      super();
      this.bitMensagem = bit;
      this.base = new BaseQuantica(base);
      this.isPerdido = isPerdido;
   }

   public MensagemQuantica(String bit, char base) {
                super();
                this.bitMensagem = bit;
                this.base = new BaseQuantica(base);
      this.isPerdido = false;
```

```
}

public BaseQuantica getBase() {
	return base;
}

public void setBase(BaseQuantica base) {
	this.base = base;
}

public String getBitBase() {
	return bitBase;
}

public void setBitBase(String bitBase) {
	this.bitBase = bitBase;
}

public String getBitMensagem() {
	return bitMensagem;
}

public void setBitMensagem(String bitMensagem) {
	this.bitMensagem = bitMensagem;
}

public boolean isIgualBase(String bit1) {
	if (bit1.equals(getBitBase())) {
		return true;
	}
	return false;
}
```

```
public boolean isIgualMensagem(String bit1) {
        if (bit1.equals(this.getBitMensagem())) {
                return true;
        }
        return false;
}

public String getSimboloBase() {
        if (getBitBase().equals("1")) {
                return "|";
        } else {
                return "-";
        }
}

public String getSimboloMensagem() {
        if (getBitMensagem().equals("1")) {
                return "|";
        } else {
                return "-";
        }
}

public MensagemQuantica(String base, String mensagem) {
        super();
        this.bitBase = base;
        this.bitMensagem = mensagem;
}

public MensagemQuantica() {
        super();
}
```

```
    public MensagemQuantica(boolean hasPerda) {
        super();
        this.isPerdido = hasPerda;
    }

            public void inverteBitBase() {
                    setBitBase(getBitBase().equals("1") ? "0" : "1");
            }

            public void inverteBitMensagem() {
                    setBitMensagem(getBitMensagem().equals("1") ? "0" : "1");
            }

            public void inverteBitAll() {
                    setBitMensagem(getBitMensagem().equals("1") ? "0" : "1");
                    setBitBase(getBitBase().equals("1") ? "0" : "1");
            }
}



/*
Classe controladora que responde a pagina index
*/
package br.ceub.scq.controle;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;

public class Resetar extends HttpServlet {
	private static final long serialVersionUID = 1L;
	private String pagDestino = "/jsp/index.jsp";

	HttpSession session;

	protected void service(HttpServletRequest req, HttpServletResponse res)
			throws ServletException, IOException {

		getSession(req).invalidate();
		RequestDispatcher red =
getServletContext().getRequestDispatcher(
				pagDestino);
		red.forward(req, res);
	}

	private HttpSession getSession(HttpServletRequest req) {
		return session!=null?session:req.getSession();
	}
}



/**
Classe controladora que responde a Configuração QKD
 */
package br.ceub.scq.controle;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;
```

```
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import br.ceub.scq.bean.BaseQuantica;
import br.ceub.scq.bean.MensagemQuantica;

/**
 * @author cid
 *
 */
public class ConfiguracaoQKD extends HttpServlet {

   private static final long serialVersionUID = 1L;
   private final String ERROR_PAGE = "/SimuladorCQ/jsp/erro.jsp";
   private RequestDispatcher red;
   private String pagDestino = "/jsp/confQkd.jsp";
   private HttpSession session;

   protected void service(HttpServletRequest req, HttpServletResponse res)
         throws ServletException, IOException {

      try {
         // Recupera parametros da JSP
         int tamanho     = Integer.parseInt(req.getParameter("tamanho"));
         int tamanhoAux  = Integer.parseInt(req.getParameter("tamanho"));
         int fase        = Integer.parseInt(req.getParameter("fase"));
         boolean          hasPerda=          req.getParameter("hasPerda").equals("0")
?false:true;
```

```
                boolean hasSpy = req.getParameter("participacao").equals("0")
?false:true;

                // Array
                ArrayList arrayAlice = new ArrayList();
                ArrayList arrayBob = new ArrayList();
                ArrayList arrayEva = new ArrayList();

                // Numero aleatório
                Random rd;

                switch (fase) {
                case 0:
                   ((HttpSession) getSession(req)).invalidate();
                   rd = new Random();
                   for (int i = 0; i < tamanho; i++) {
                      arrayAlice.add(new
MensagemQuantica(String.valueOf(rd.nextInt(2)%2),rd.nextInt(2)%2==0?'A':'B'));
                   }
                   getSession(req).setAttribute("arrayAlice", arrayAlice);
                   break;

                case 1:
                   rd = new Random();
                   boolean hasPerdaAux = false;
                   for (int i = 0; i < tamanho; i++) {
                      if(hasPerda){
                         hasPerdaAux = rd.nextInt(20)%4!=0?false:true;
                         System.out.println("Bit " + i + " foi perdido? " + hasPerdaAux);
                      }
                      arrayBob.add(new
MensagemQuantica(String.valueOf(rd.nextInt(4)%2),rd.nextInt(4)%2==0?'A':'B',
hasPerdaAux));
```

```
                    arrayEva.add(new
MensagemQuantica(String.valueOf(rd.nextInt(4)%2),rd.nextInt(4)%2==0?'A':'B',
hasPerdaAux));
                }

                if (!hasSpy) {
                    arrayEva = new ArrayList();
                }

                getSession(req).setAttribute("arrayBob", arrayBob);
                getSession(req).setAttribute("arrayEva", arrayEva);
                break;

            case 2:
                String str = "";
                arrayAlice = (ArrayList) getSession(req).getAttribute("arrayAlice");
                arrayBob = (ArrayList) getSession(req).getAttribute("arrayBob");

                if (!hasSpy) {
                    for (int i = 0; i < tamanho; i++) {
                        MensagemQuantica beanAlice = (MensagemQuantica)
arrayAlice.get(i);
                        MensagemQuantica beanBob = (MensagemQuantica)
arrayBob.get(i);

                        BaseQuantica baseAlice = (BaseQuantica) beanAlice.getBase();
                        BaseQuantica baseBob = (BaseQuantica) beanBob.getBase();

                        if(beanBob.isPerdido()){
                            str += "P";
                        } else {
                            if (baseAlice.isIgualBase(baseBob.getTipoBase())) {
                                str += beanAlice.getBitMensagem();
```

```
                        } else {
                            str += "#";
                        }
                    }
                }
            } else {
                arrayEva = (ArrayList) getSession(req).getAttribute("arrayEva");
                for(int i = 0; i < tamanho; i++){
                    MensagemQuantica beanAlice = (MensagemQuantica)
arrayAlice.get(i);
                    MensagemQuantica beanBob = (MensagemQuantica)
arrayBob.get(i);
                    MensagemQuantica beanEva = (MensagemQuantica)
arrayEva.get(i);

                    BaseQuantica baseAlice = (BaseQuantica) beanAlice.getBase();
                    BaseQuantica baseBob = (BaseQuantica) beanBob.getBase();
                    BaseQuantica baseEva = (BaseQuantica) beanEva.getBase();

                    if(beanBob.isPerdido()){
                        str += "P";
                        tamanhoAux--;
                    } else {
                        if(baseAlice.isIgualBase(baseEva.getTipoBase())){
                            if (baseAlice.isIgualBase(baseBob.getTipoBase())) {
                                str += beanAlice.getBitMensagem();
                            } else {
                                str += "#";
                                tamanhoAux--;
                            }
                        }else{
                            rd = new Random();
                            if (baseAlice.isIgualBase(baseBob.getTipoBase())) {
```

```
                        int auxRd = rd.nextInt()%2;
                        auxRd = auxRd<0?auxRd*-1:auxRd;
                        str += auxRd;
                    } else {
                        str += "#";
                        tamanhoAux--;
                    }
                }
            }
        }
        getSession(req).setAttribute("str", str);

        String strComparada = "";
        int contadorErro = 0;
        int contadorPerda = 0;
        int contadorBaseIncorreta = 0;
        for (int j = 0; j < tamanho / 2; j++) {
            char valorBit = str.charAt(j);
            if (valorBit == '#') {
                contadorBaseIncorreta++;
            } else if(valorBit == 'P'){
                contadorPerda++;
            } else {
                MensagemQuantica beanAlice = (MensagemQuantica)
arrayAlice.get(j);
                if (valorBit == beanAlice.getBitMensagem().charAt(0)) {
                    strComparada += valorBit;
                } else {
                    strComparada += "E";
                    contadorErro++;
                }
            }
```

```
                    if(tamanhoAux == j)
                        break;
                }
                getSession(req).setAttribute("strComparada", strComparada);
                getSession(req).setAttribute("contadorErro",
String.valueOf(contadorErro));
                getSession(req).setAttribute("contadorPerda",
String.valueOf(contadorPerda));
                getSession(req).setAttribute("contadorBaseIncorreta",
String.valueOf(contadorBaseIncorreta));
                break;

            case 3:

                break;

            default:
                break;
            }

            red = getServletContext().getRequestDispatcher(pagDestino);
            red.forward(req, res);

        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(e.getMessage());
            res.sendRedirect(ERROR_PAGE);
        }
    }

    private HttpSession getSession(HttpServletRequest req) {
        return session != null ? session : req.getSession();
    }
```

}